MSAT-X
Report 125

P-141

General Electric Corporate Research and Development
Communication Technology and Systems Branch
Schenectady, New York 12345

(NASA-CR-182377) AN ADDITIONAL STUDY AND IMPLEMENTATION OF TONE CALIBRATED TECHNIQUE OF MODULATION Final Report (Jet Propulsion Lab.) 141 p CSCL 17B

N88-15117

Unclas
G3/32 0117210

AN ADDITIONAL STUDY AND IMPLEMENTATION OF TONE CALIBRATED TECHNIQUE OF MODULATION

IN-32
117210
DATE OVERRIDE

Final Report

Authors:

Dr. W. Rafferty
L. K. Bechtel
N. E. Lay

November 1985

JPL Contract Number 957190

This report was prepared for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration.

CONTENTS

1. INTRODUCTION

The material presented in this Final report is concerned with the work performed on An Additional Study and Implementation of Tone Calibrated Technique of Modulation for the Jet Propulsion Laboratory, contract number 957190.

Earlier studies [1,2,3] have shown that pilot-aided coherent modulation techniques, of which the Tone Calibrated Technique (TCT) is an example, can be used to alleviate the effects of multipath fading in Land Mobile Radio (LMR) communication links. It appears, at the moment, that there are no commercial pilot-aided LMR links in operation. Conventional LMR links predominantly use non-coherent detection methods at the receiver to allow rapid burst signal acquisition. This also has the advantage of fast re-acquisition of the the transmit signal when multipath fading is encountered. However, non-coherent detection generally has a poorer bit error rate (BER) performance than coherent and differentially coherent detection schemes.

Coherent detection schemes which regenerate a phase reference directly from the received signal, are typically not used on LMR links due to the complexity involved in configuring them to track fading signals. As an alternative, pilot-aided techniques have much to offer due to the presence of a transmitted reference. They are the subject of this report. The differentially coherent receiver enjoys a BER performance which, for additive white gaussian channels, lies somewhere between that of the coherent and non-coherent detection schemes. In the presence of multipath fading, this receiver displays an irreducible error detection floor, just as the in the case of the non-coherent detector.

The TCT communication method has been shown [4] to be theoretically free from an error floor, and is only limited, in practice, by implementation constraints. Section 2 of this report introduces the concept of the TCT transmission scheme along with a baseband implementation of a suitable demodulator. Two techniques for the generation of the TCT signal are considered: a Manchester source encoding scheme (MTCT) and a subcarrier based technique (STCT).

Section 3 summarizes the results of the TCT link computer simulation which was previously described in detail in the First Interim Report[5]. Section 4 addresses the hardware implementation of the MTCT system and outlines the digital signal processing design considerations involved in satisfying the modulator/demodulator requirements.

Section 5 presents a discussion on the program findings and suggests future directions based on conclusions made regarding the suitability of the TCT system for the transmission channel presently under consideration.

2. TCT DATA TRANSMISSION SYSTEM

The Tone Calibrated Technique has been explored in a related program [1], sponsored by JPL, and discussed in the open literature [2]. The underlying concept is the use of a pilot, or tone, to synchronously remove channel perturbations from the data bearing sidebands and, simultaneously, perform coherent data detection. This requires that the pilot be situated at a suitable location in the transmit frequency band; the channel distortions should be distributed symmetrically around this location and there should be no data sideband energy present. It also assumes that the transmit link is linear to avoid capture of the pilot due to hardlimiting effects.

The TCT system positions the pilot at the midpoint of the transmit band. Prior to carrier modulation, the data modulated signal has been removed from this location by the creation of a spectral null at zero frequency. The data modulation scheme employed is M-ary Phase Shift Keying (PSK); this requirement is imposed by channel bandwidth constraints and is not a limitation of the TCT method, which is compatible with many other types of modulation schemes. For the given signalling requirements, the transmission of data at a rate of 2.4 kbps in a 3.6 kHz RF bandwidth, using quadriphase PSK (QPSK) modulation offers the minimum possible PSK signalling set.

The previous TCT program [1] was concerned with the proof-of-concept phase and was principally a harware implementation study. Of major interest was the realization of the TCT demodulator and its performance in the presence of additive white gaussian noise (AWGN) and hardware simulated multipath fading. An IF TCT demodulator was designed and studied. Results derived from the experimental set-up indicated that the pilot-aided demodulation could remove the error floor normally associated with non-coherent and strictly coherent receivers, the latter case holding true when conditioned on the fact that no attempt has been made to compensate for fading effects. This result was subsequently supported by theoretical analyses [4]. The TCT system bit error rate performance appears to be acceptable given the power penalty of including the pilot. This encouraging result was somewhat overshadowed by the doubling of the transmit bandwidth over that actually required by other modulation schemes. This is due to the Manchester source encoding which was employed to generate the spectral null around zero frequency.

Although the bandwidth constraints can be mitigated by using higher order PSK signalling sets, it is reasonable to assume that their use would cause the link bit error rate performance to degrade significantly due to the reduced decision space and sensitivity to the recovered pilot.

This report is concerned with two major areas of the TCT system, (a) the generation of the pre-modulation spectral null in the transmitter and (b) the use of baseband processing techniques to implement the demodulator. The use of digital signal processing elements to implement both the modulator and demodulator was considered to be a key element and, consequently, is a topic of detailed discussion in the report.

Two schemes were initially investigated for the generation of the TCT signalling format and are described in the remainder of this section. The first to be considered is based on a Manchester encoding scheme to generate the spectral null at the transmitter; this was the method investigated in the previous study. The other method considered centers on the use of subcarrier modulation techniques for spectral null generation. It was decided during the course of the program, primarily due to time constraints, to pursue only the Manchester TCT system to full hardware implementation.

2.1 Manchester Encoded TCT

2.1.1 Modulator

Manchester source encoding acts upon the input digital data in such a way as to redistribute the data energy away from zero frequency. The resulting power spectrum is given by:

$$M(f) = \sin^2(x)\,\mathrm{sinc}^2(x) \tag{2.1}$$

where

$$x = \pi f T_s/2$$

$$\mathrm{sinc}(x) = \sin(x)/x$$

Eqn. (2.1) clearly shows that a null is created at zero frequency and that the main spectral lobe has almost doubled in width. To meet the single-sided RF occupancy requirements, which call for the transmit signal to be attenuated by 40dBC at 1.8 kHz removed from the center of the band, some form of spectral shaping must be used. The shaping employed is the raised-cosine pulse in the frequency domain with a maximum excess bandwidth fraction, β, of 0.5. The shaping is implemented as a time domain function and can be expressed as follows:

$$p(t) = \frac{\sin(\pi t/T_b)}{\pi t/T_b} \; \frac{\cos(\pi\beta t/T_b)}{(1-(2\beta t/T_b)^2)} \qquad -\infty < t < \infty \tag{2.2}$$

where T_b is the bit rate, 2.4 kbps. The corresponding frequency spectrum is given by eqn.(2.3).

$$p(f) = \begin{cases} 1; & 0 \le f \le (1-\beta)/2T_b \\ \tfrac{1}{2}[1-\sin(\pi T_b(f-1/2T_b)/\beta)]; & \dfrac{(1-\beta)}{2T_b} \le f \le \dfrac{(1+\beta)}{2T_b} \end{cases} \tag{2.3}$$

As previously mentioned, the modulation scheme employed is QPSK, so the input data is split into even and odd streams, where the Manchester encoding and pulse-shaping will be performed independently of each other. Figure 2.1 illustrates the complete Manchester system pre-modulation data processing.

Included in both processing paths of Figure 2.1 are highpass filters to enlarge the necessary spectral null. It has been determined [1] that Manchester encoding by itself does not create a sufficiently wide spectral null. There is simply too much residual data energy which will overlap the pilot recovery passband at the receiver and result in the degradation of the TCT calibration process in the demodulator. Moreover, the pulse-shaping employed has a constant amplitude frequency response in the vicinity of d.c. (see 2.3), hence the need for additional spectral shaping through the use of the highpass filters.

The pre-modulation processing is implemented digitally, this allows for the use of linear phase, finite impulse response highpass filters. In this way, an attempt is made to minimize the effects of intersymbol interference (ISI) arising from the removal of the low frequency data energy.

The output of the highpass filters constitute the inphase (I) and quadrature (Q) components of the QPSK modulation. It is desired to use I and Q path staggering to generate offset QPSK (OQPSK) as this reduces transmit envelope amplitude variations. The Q path, Sq(t), is therefore delayed $T_b/2$ seconds relative to the I path, Si(t). The exact description of Si(t) and Sq(t) is delayed until section 4.2 where they are described in relation to the hardware implementation. Signals Si(t) and Sq(t) are converted to analog waveforms, then passed through lowpass reconstruction filters to generate signals Si'(t) and Sq'(t). These are then used to modulate a quadrature carrier pair as follows,

$$\mathrm{OQPSK}(t) = Si'(t)\cos(w_0 t) + Sq'(t)\sin(w_0 t) \qquad (2.4)$$

where w_0 is the radian carrier frequency.

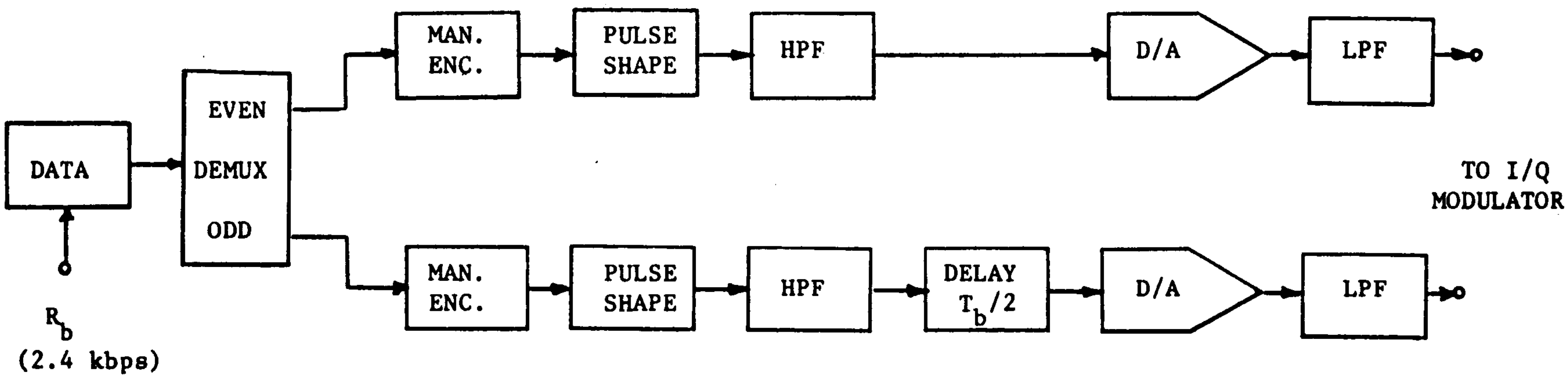


FIGURE 2.1 TCT MANCHESTER BASEBAND PROCESSING

2.1.2 Demodulator

The demodulator configuration of the previous TCT study was an IF arrangement which employed analog techniques. This study is considered to be the next stage in the TCT demodulator development since it employs digital baseband I/Q signal processing. The baseband approach is particularly attractive because its low data rate makes possible the extensive use of various DSP techniques. Many signal processing chips are commercially available, and could be utilized in this low data rate environment. This should reduce the complexity of TCT transceiver hardware as well as result in an implementation that is well suited to custom IC fabrication.

The demodulator configuration investigated was derived from that suggested by Davarian [2]. Figure 2.2 illustrates the digital Manchester based TCT demodulator.

The received signal at the demodulator consists of the pilot tone and OQPSK data signal, corrupted by multipath fading, along with a thermal noise component. This signal can be expressed as follows,

$$\begin{aligned} r(t) = {} & aX_t\cos(w_0t + Y_t) \\ & + A/\sqrt{2}\ Si(t)W_t\cos(w_0t + Y_t) \\ & + A/\sqrt{2}\ Sq(t)X_t\sin(w_0t + Y_t) \\ & + Ni(t)\cos(w_0t) + Nq(t)\sin(w_0t) \end{aligned} \tag{2.5}$$

The first term of eqn.(2.5) represents the the pilot term, the second and third terms correspond to the OQPSK signal, while the remaining terms are attributed to the thermal noise. X_t and Y_t are random variables which describe the amplitude and phase variations used to model the multipath fading effects [2].

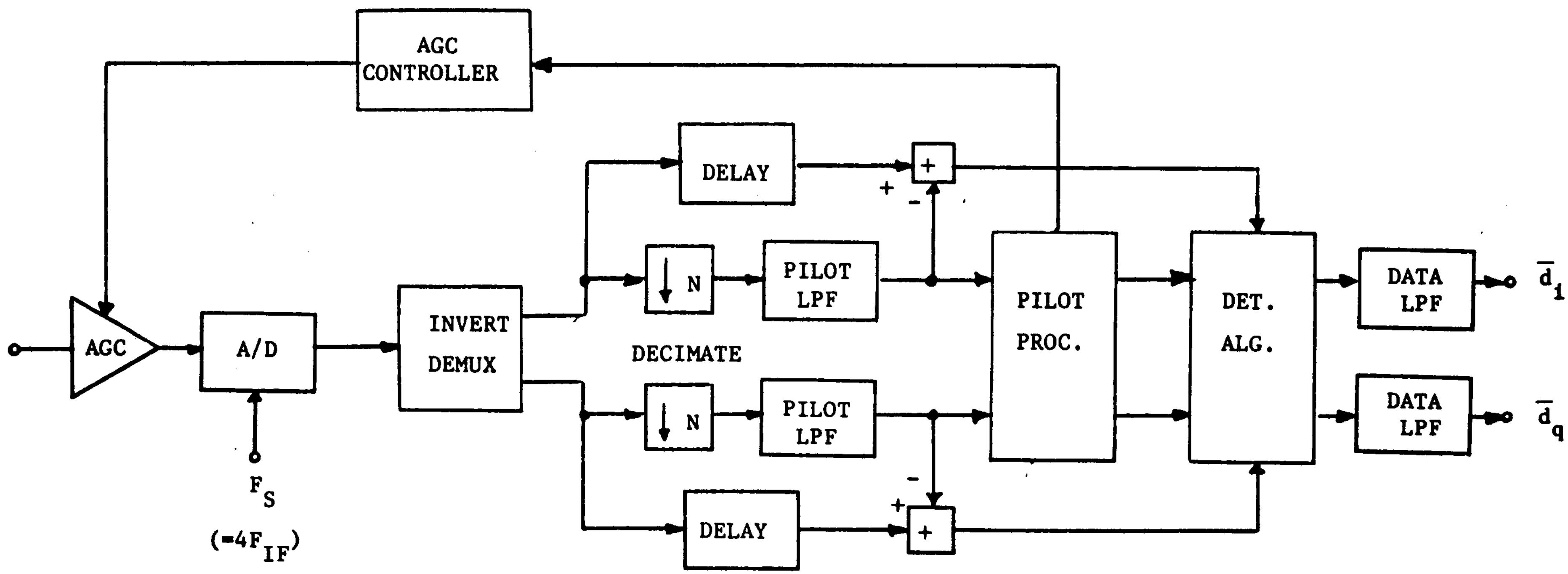


FIGURE 2.2 DIGITAL MTCT DEMODULATOR

The received signal is first passed through a bandpass filter and the AGC amplifier before it is translated to baseband, where the majority of the processing will take place. The bandpass filter is used to reduce the composite input signal strength and excess noise bandwidth. The translation to baseband is performed by mixing the received signal with quadrature sinusoids operating at the IF frequency. In this way the inphase (I) and quadrature (Q) signals necessary for baseband processing are generated.

The I and Q components of the received pilot are recovered by passing the two streams through a filter which has a passband equivalent to the fading bandwidth [1]. In parallel with this operation, the I and Q signals are also fed to a delay buffer which compensates for the overall pilot processing time. The delayed signals will eventually be utilized in conjunction with the appropriately processed pilot I and Q signals, to obtain an estimate of the transmitted data.

The received signal is corrupted by channel perturbations and for the link considered, these perturbations are induced by additive white gaussian noise and multipath fading. Hence, the recovered pilot signal will have impressed upon it amplitude and phase related information about the fading. The task of the pilot processing section is to extract this information from the recovered pilot prior to data detection. A linear TCT pilot processor that was considered previously produces an output with an amplitude component which is the reciprocal of the input pilot amplitude. This method necessitates a squaring and division operation. The processing scheme employed here uses I/Q hardlimiting in a similar fashion to that proposed earlier by Davarian[6]. This has two advantages: it removes the fading amplitude component from the recovered pilot and it simplifies the arithmetic processing requirements. The I/Q hardlimiting is performed by taking the arctangent of Q_p/I_p, recall (2.5),

$$\Phi = \tan^{-1}\left[\frac{aX_t\cos(\Omega t + \theta_0 - Y_t)}{aX_t\sin(\Omega t + \theta_0 - Y_t)}\right] = (\Omega t + \theta_0 - Y_t) \tag{2.6}$$

where θ_0 and Ωt are residual system phase and frequency offsets. It can be seen that the amplitude fading has been removed and that the output of the arctangent function is an estimate of the phase perturbation process. This output is then passed on to the detection algorithm. Since data detection is a phase comparison process the pilot amplitude is not strictly required. The long term data sideband amplitude variations can be handled using an AGC loop with a control signal as given below. Faster variations can be addressed by maintaining a sufficient processing signal-to-noise ratio to meet the system performance requirements at low absolute input levels, however, this is probably only realistic for Rician fading channels.

The detection section of the demodulator performs the simultaneous operations of data recovery and the removal of channel phase perturbations. The ouput of the arctangent function is converted to sine and cosine terms which act as coherent phase references. The detection algorithm can be expressed as follows,

$$Z_I = I_D C + Q_D S \tag{2.7a}$$

$$Z_Q = Q_D C - I_D S \tag{2.7b}$$

where

$$C = 2\cos(\Phi), \quad S = 2\sin(\Phi)$$

and Φ is as given in 2.6. I_D and Q_D are the outputs of the delay buffers from which the inphase and quadrature components of the recovered pilot, I_p and Q_p, have been removed, see Figure 2.2. This is accomplished by the action of the adders immediately following the delay buffers. Signals Z_I and Z_Q are then passed on to integrate-and-dump filters to produce estimates of the transmitted data.

An AGC control signal can be derived by taking advantage of the pilot and data detection processors. For example, the amplitude variations on the inphase pilot component can be obtained as follows.

$$E = I_p/\cos(\Phi) \tag{2.8}$$

E can then be compared to a nominal value and suitably lowpass filtered to generate a control signal which will be used to set the gain of the IF AGC amplifier.

2.2 Subcarrier TCT

The subcarrier version of the TCT (STCT) modem is, in fact, very similar to the previously described Manchester modem. The main difference between the two is the manner in which the spectral null at d.c., which is necessary for the proper transmission of the pilot tone, is created. The MTCT version relies upon Manchester coding followed by highpass filtering of the shaped data to remove unwanted sideband energy around zero frequency. The STCT method, on the other hand, modulates the shaped data onto a very low frequency subcarrier to redistribute its data sideband energy away from d.c. The resulting frequency spectra of the two methods are similar, as would be expected, however, software simulations indicate that the subcarrier method does provide some advantages, and, hence, warrants attention.

2.2.1 Modulator

Two advantages of the subcarrier method are immediately obvious. For one, the modulation allows the arbitrary location of the data sidebands in a symmetric position around d.c. Also, the pulse shaping can then be used to control the low and high-side roll-off of these sidebands, which cannot be done in the case of the Manchester encoded data.

The implementation of this STCT modulator is quite simple, see Figure 2.3. The data bits are split into even and odd streams, bipolar encoded, then pulse-shaped using the same raised-cosine filtering employed in the MTCT modulator. In this case, however, the excess bandwidth fraction must be reduced to 0.4. The shaped data streams then modulate a quadrature subcarrier signal pair operating at a frequency of 960 Hz. This frequency is a submultiple of the data clock, which is very useful from an implementation standpoint since

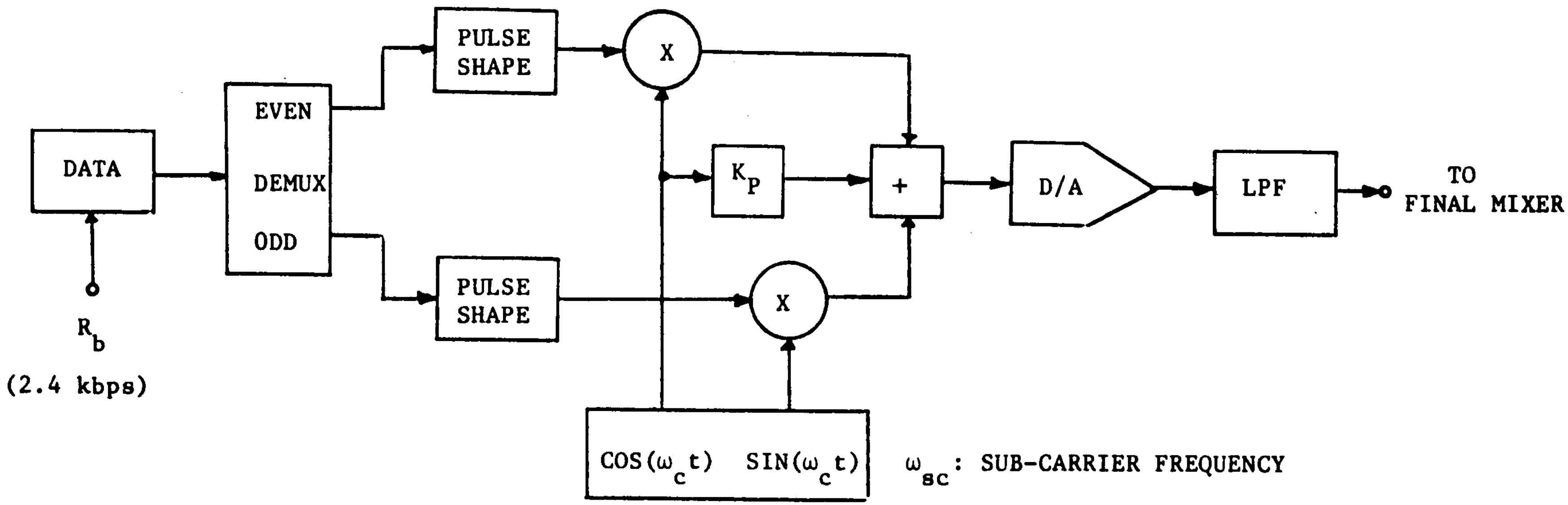


FIGURE 2.3 SUB-CARRIER TCT MODULATOR

the sine and cosine generators are effectively synchronized to this data clock.

It will be shown in section 3.2.1 that this method of frequency redistribution does in fact produce the desired null at d.c. without the need for dual highpass filters and the attendant penalty of added data ISI.

2.2.2 Demodulator

The demodulator for the subcarrier version of TCT, illustrated in Figure 2.4, resembles its MTCT counterpart. The difference here is the added remodulation and subcarrier phase estimation functions. Fading compensation and sychronous detection are performed at the subcarrier frequency; hence the need for the remodulation function which modulates the recovered pilot phase onto locally generated sine and cosine subcarrier phase references. These local references are produced by phase-locking a local source to the incoming suppressed subcarrier. It should be noted that the action of the pilot in the synchronous detector is to remove any residual frequency and phase offsets, thus the subcarrier recovery circuits, in theory, only have to align the phase of the local reference to the transmitted one.

This synchronization is performed by a first order phase-locked loop. The outputs of the final data filters serve as a source of the receive phase states. The receive phase states are compared to the known nominal receive data phase states. The angular difference between these two quantities becomes the error signal. The phase-locked loop output is an estimate of the phase difference between the local and the received subcarrier, this is added to the recovered pilot phase angle prior to the remodulation process.

3. TCT SYSTEM COMPUTER SIMULATION

The initial investigation of the baseband Tone Calibrated Technique required the development of a software simulation. A simulation was generated which omitted the modelling of any channel perturbations so that the modem performance could be evaluated solely in terms of system parameters.

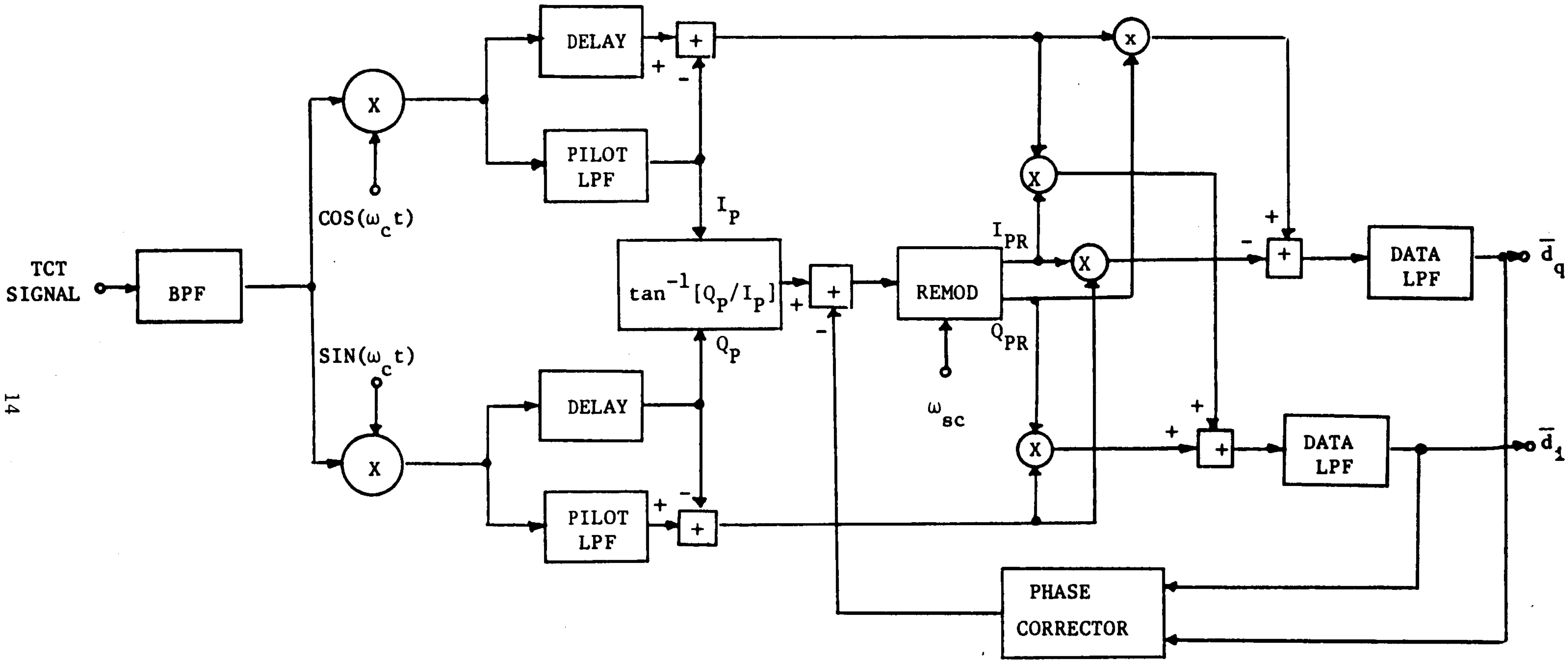


FIGURE 2.4 STCT DEMODULATOR

Subsequent development continued to focus on the signal processing structures of the modem. By concentrating on emulating the elements of the real-time implementation, the simulation effectively became a valuable design tool. The structure of the final design could be fully developed before committing it to hardware, allowing for a relatively smooth implementation. The inclusion of multipath fading and thermal noise to the simulation was not viewed as critical, since the impact of these effects would be examined in the real-time system.

The simulations were written in FORTRAN and interfaced to the Interactive Laboratory System (ILS) digital signal processing software package for purposes of graphical presentation and analysis. In addition, ILS was employed in designing all the required digital filters. Linear phase filters were used throughout to minimize phase distortion and the design algorithm employed was the Remez Exchange Algorithm which produces equiripple in both the passband and stopband. A data eye pattern was generated from a $2^{10} - 1$ pseudo-random bit sequence and the recovered eye quality was used as an indication of system performance. ILS is commercially available through Signal Technology, Inc.

Appendix I contains a program listing of the MTCT modem simulation described in the next section.

3.1 Manchester Encoded TCT

3.1.1 Modulator

The Manchester encoded TCT modulator was first simulated as shown in Figure 2.1, except without the highpass filters. Since no channel modelling was performed, the data was QPSK modulated to 12 kHz, the receiver final IF frequency. The data at IF was represented as a 48 kHz sampled signal, consistent with the demodulator IF sampling frequency.

In order to meet the specified bandwidth requirements, frequency domain raised-cosine pulse-shaping was employed with a Nyquist excess fraction, β, of 0.5 at a data rate of 2.4 kbps. Figures 3.1(a) and 3.1(b) show respectively the baseband data eye and the data eye spectrum. The pulse shape extends over

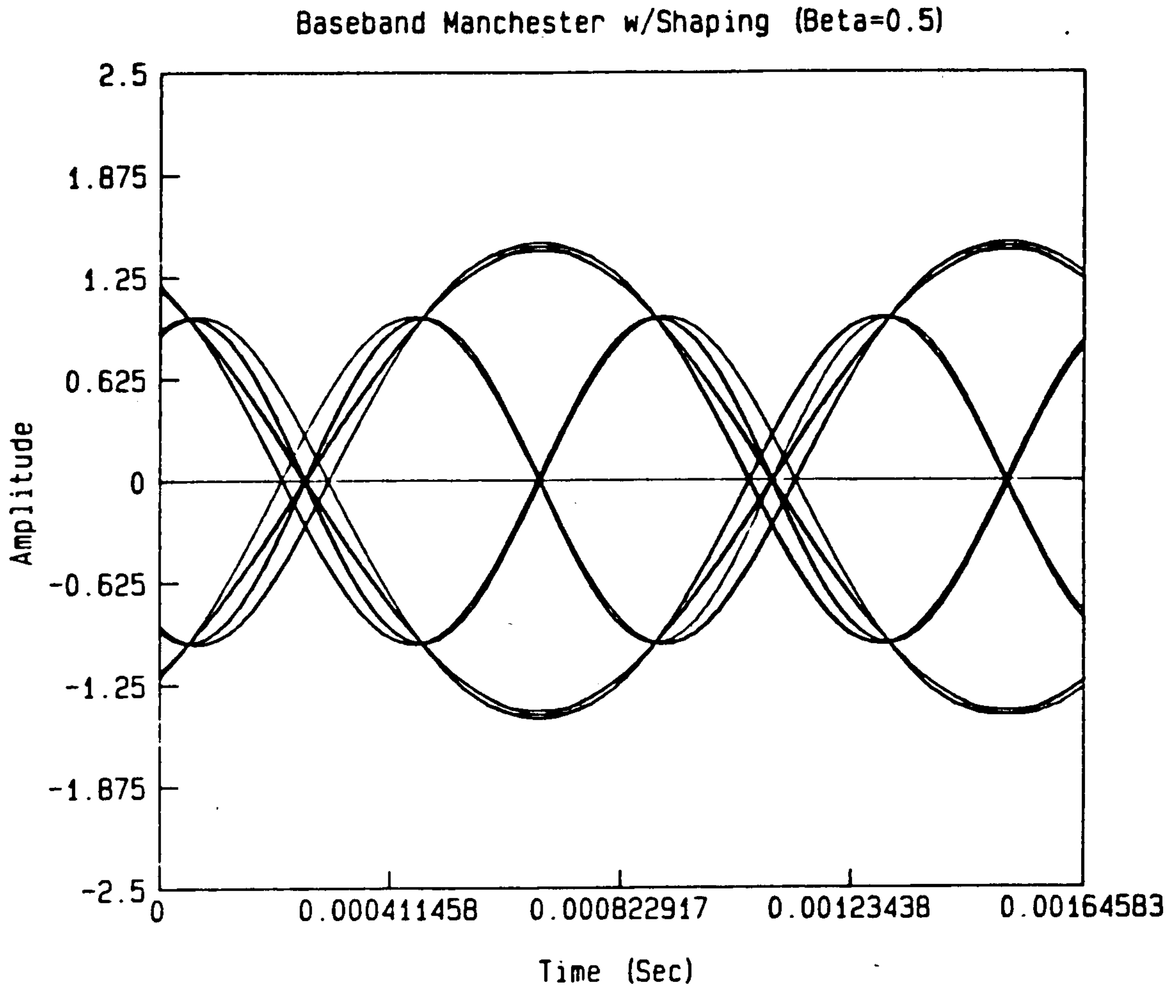


FIGURE 3.1(a) SHAPED MANCHESTER ENCODED DATA, NO HIGHPASS FILTER

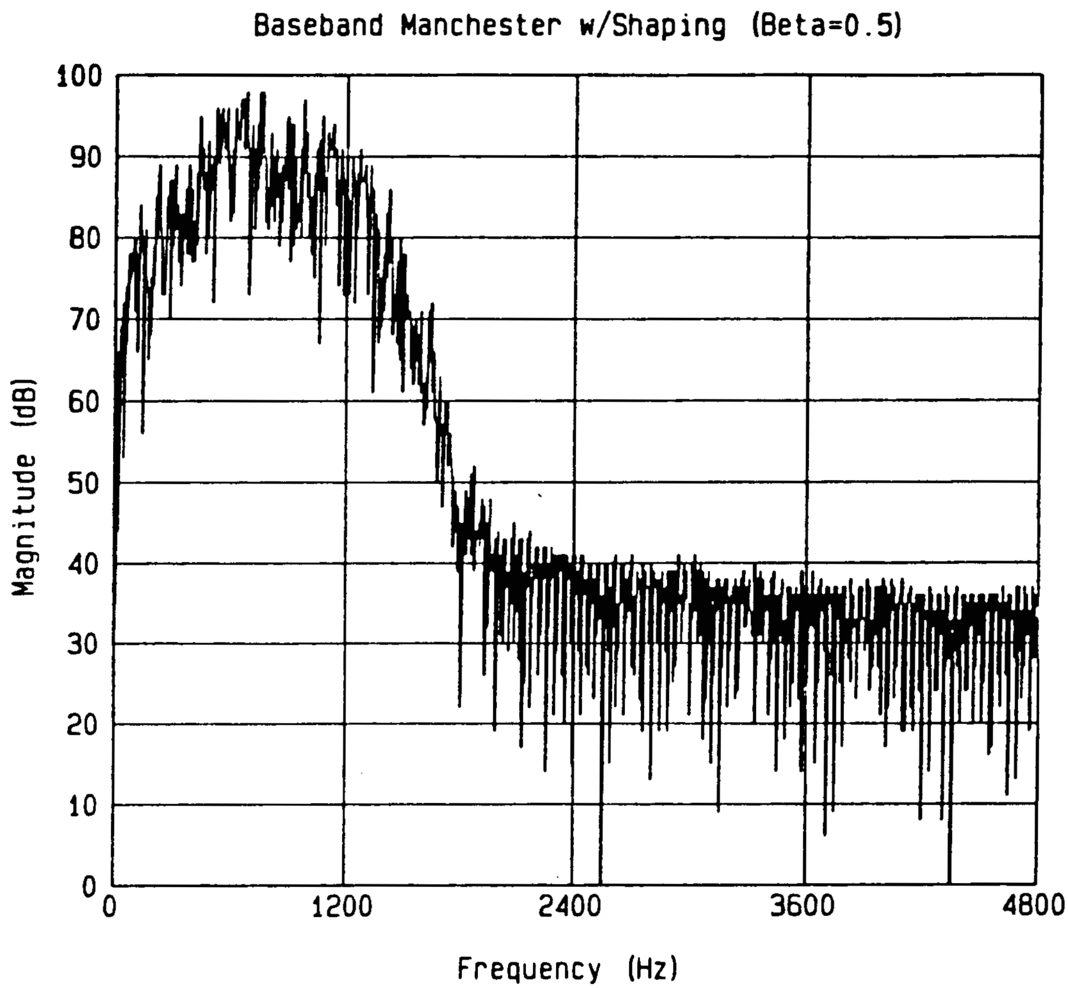


FIGURE 3.1(b) FREQUENCY SPECTRUM OF FIGURE 3.1(a)

eight Manchester bit time periods and provides sufficient spectral occupancy characteristics, i.e. over 40 dB of attenuation at 1.8 kHz from the center frequency, while generating an eye pattern with points of no ISI.

Although the pulse shaping is more than adequate in fulfilling bandwidth constraints, the generation of a zero frequency null is poorly implemented with Manchester encoding alone. A null width of 100 Hz is desired for the insertion of a carrier to track multipath fading characteristics. If a highpass filter is employed, a significant null can be created, see Figure 3.2(b); however, Figure 3.2(a) shows the distortion introduced by filtering out low frequency components from the data.

A 91 tap highpass filter is used initially as a control to establish a frequency response with a very sharp cut-off and no passband ripple. The 91 tap filter has a 3 dB point at 150 Hz and functions at the baseband processing rate of 9.6 kHz. The degradation due to this filter is approximately 16% as measured by comparing the size of the eye closure to the nominal opening. A 45 tap highpass filter design is also examined, since the longer length filter cannot be implemented in the proposed processor for the real-time system. The frequency response of this filter has a half dB ripple in the passband, a 3 dB point at 150 Hz and less attenuation in the stopband. Figures 3.3(a) and 3.3(b) show the data eye pattern and the data frequency spectrum. Both diagrams appear quite close to the 91 tap control filter simulation results, with the 45 tap filter eye pattern showing slightly more ISI.

The final step in the modulator simulation involved generating a 48 kHz sampled signal from the 9.6 kHz baseband process for the demodulator IF input requirements. This was done in two steps. First, the 9.6 kHz signal was zero insertion upsampled by a factor of 1:5 to a sampling frequency of 48 kHz. The spectrum of this signal is shown in Figure 3.4(a). The baseband spectrum is centered at multiples of 9.6 kHz. The high frequency images are then removed by lowpass filtering, yielding the spectrum of Figure 3.4(b). Although the baseband images are still recognizable, they represent negligible energy compared to the data spectrum. The 48 kHz sampled baseband data is then QPSK modulated to the 12 kHz demodulator IF frequency.

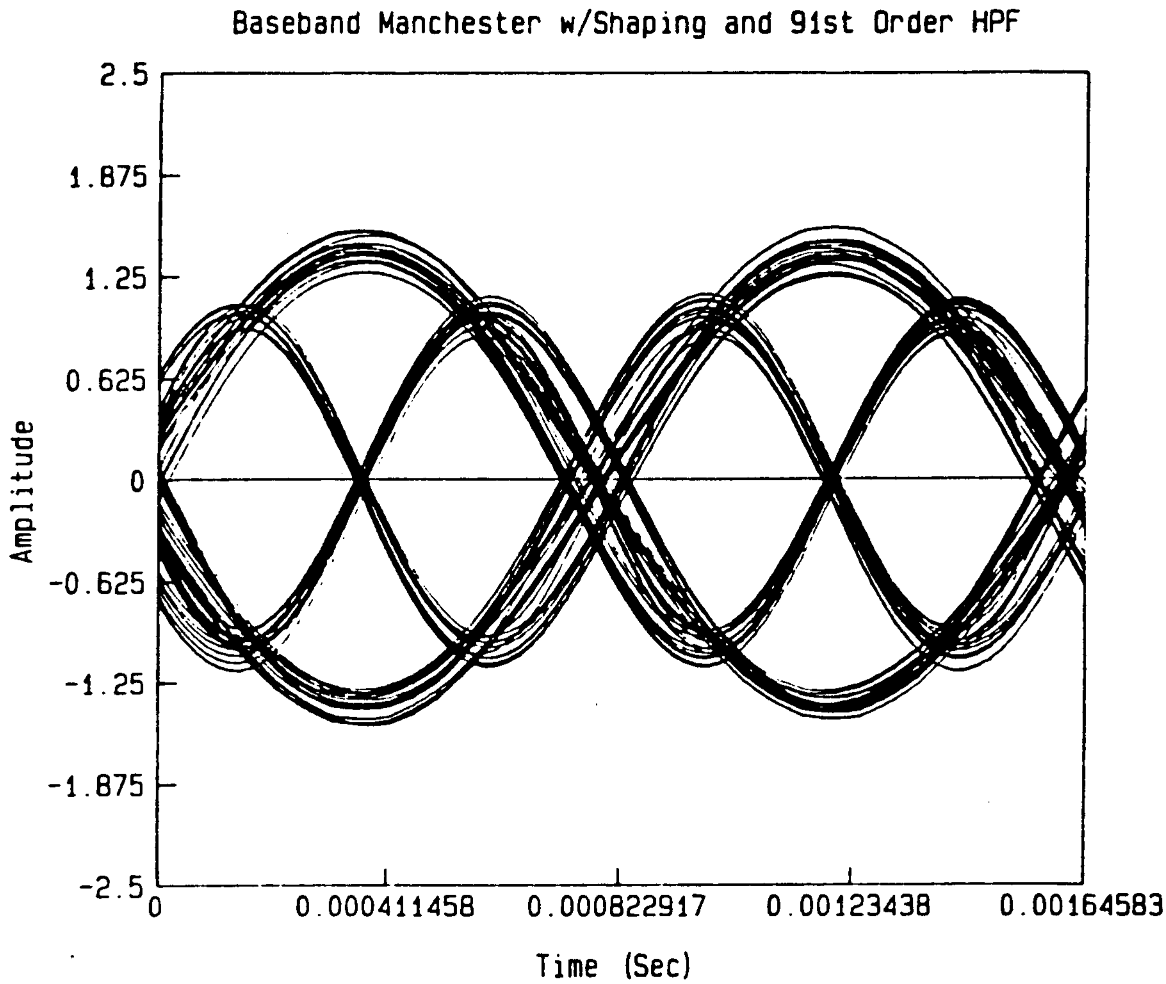


FIGURE 3.2(a) SHAPED MANCHESTER ENCODED DATA WITH HIGHPASS FILTERING (91st ORDER)

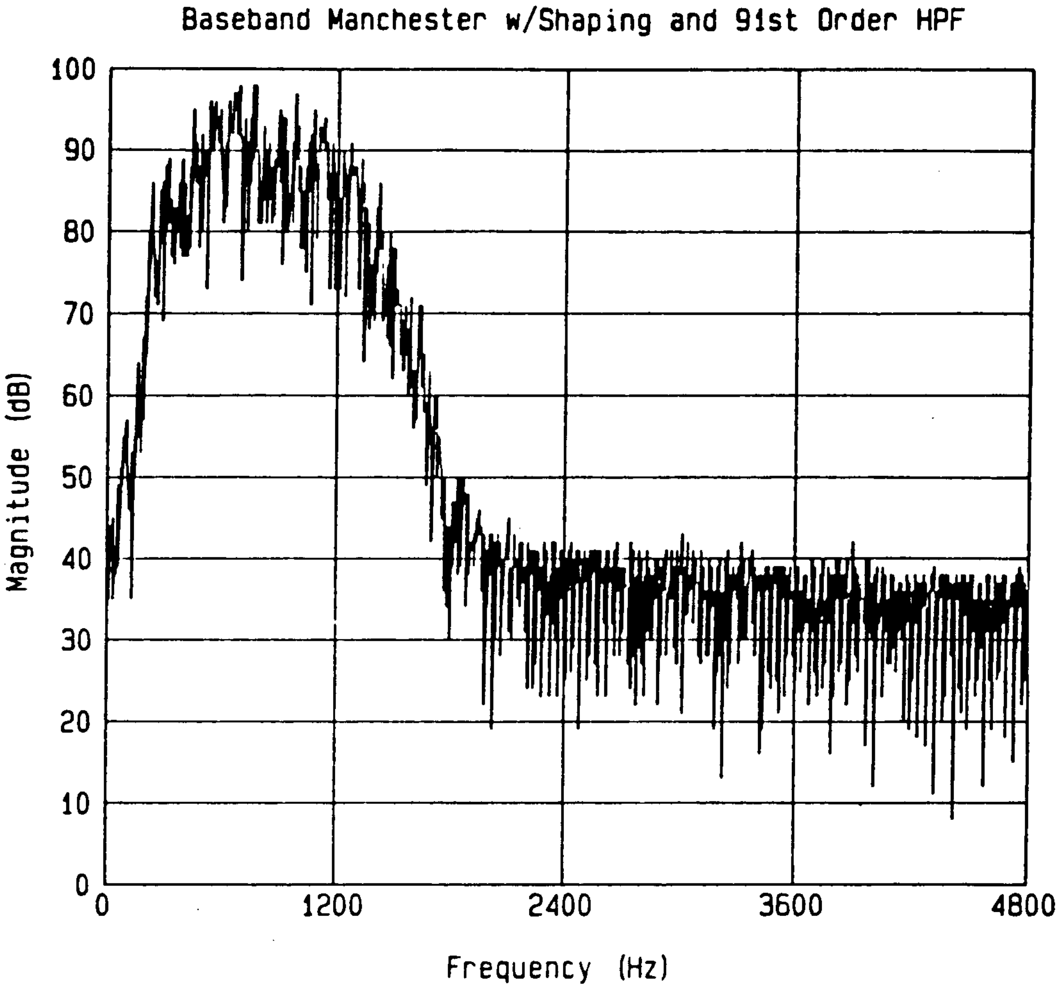


FIGURE 3.2(b) FREQUENCY SPECTRUM OF FIGURE 3.2(a)

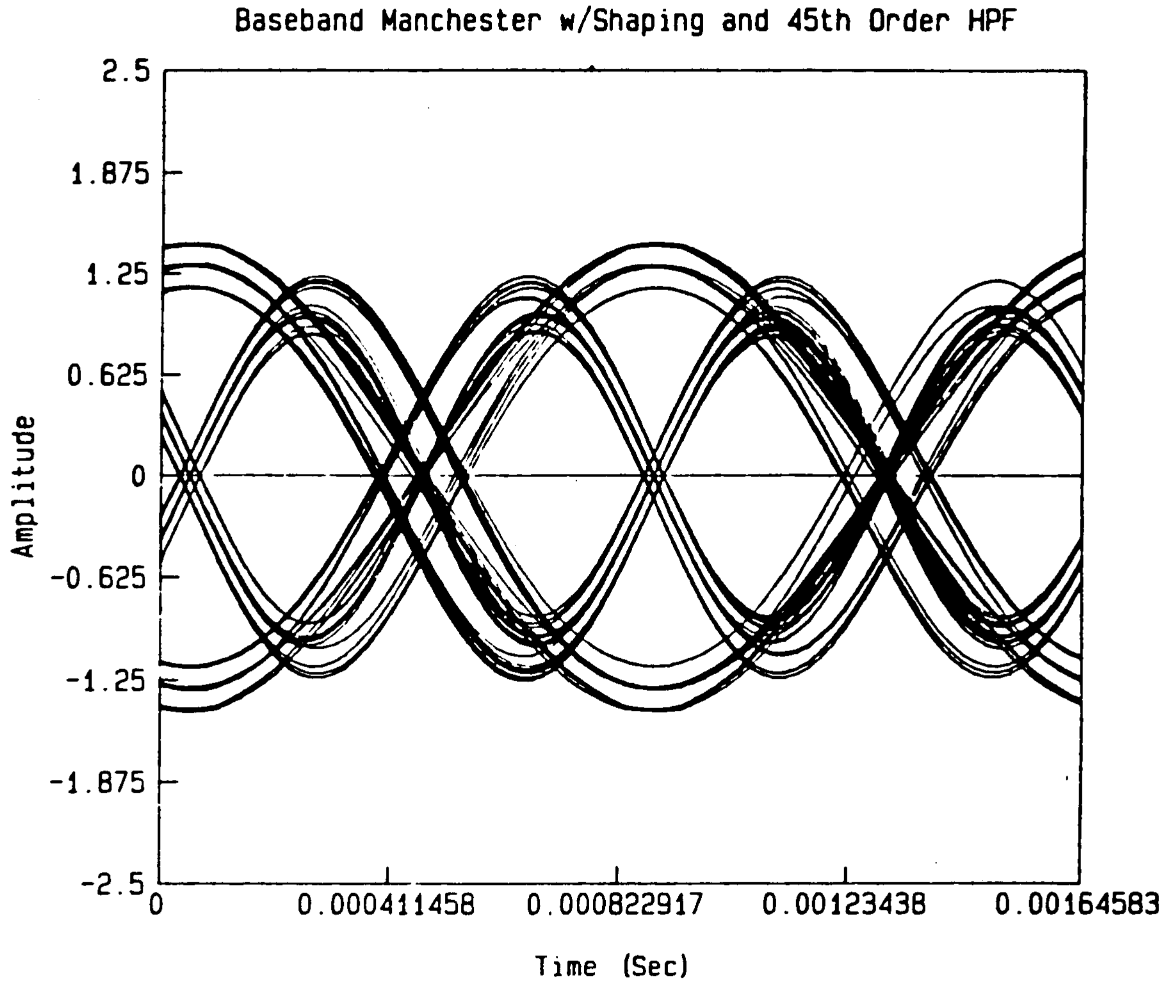


FIGURE 3.3(a) SHAPED MANCHESTER ENCODED DATA WITH HIGHPASS FILTERING (45th ORDER)

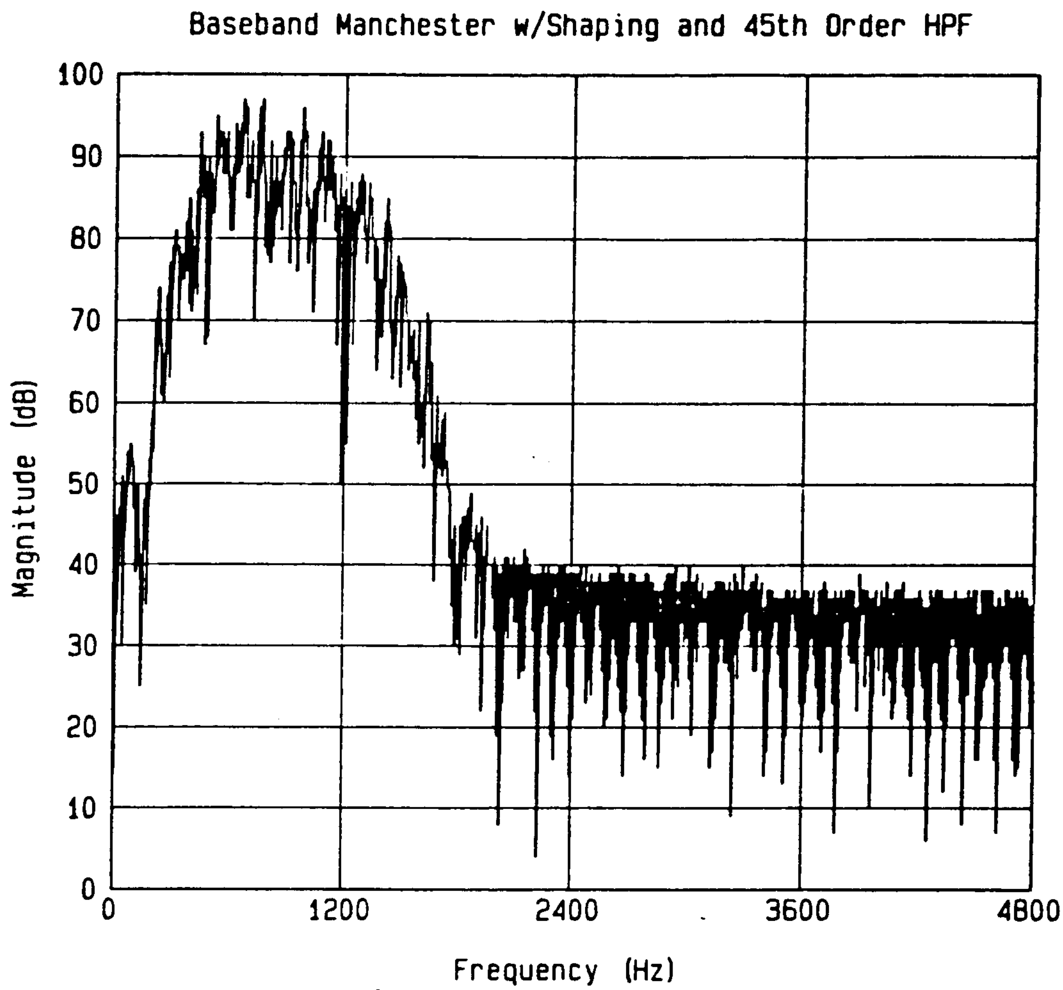


FIGURE 3.3(b) FREQUENCY SPECTRUM OF FIGURE 3.3(a)

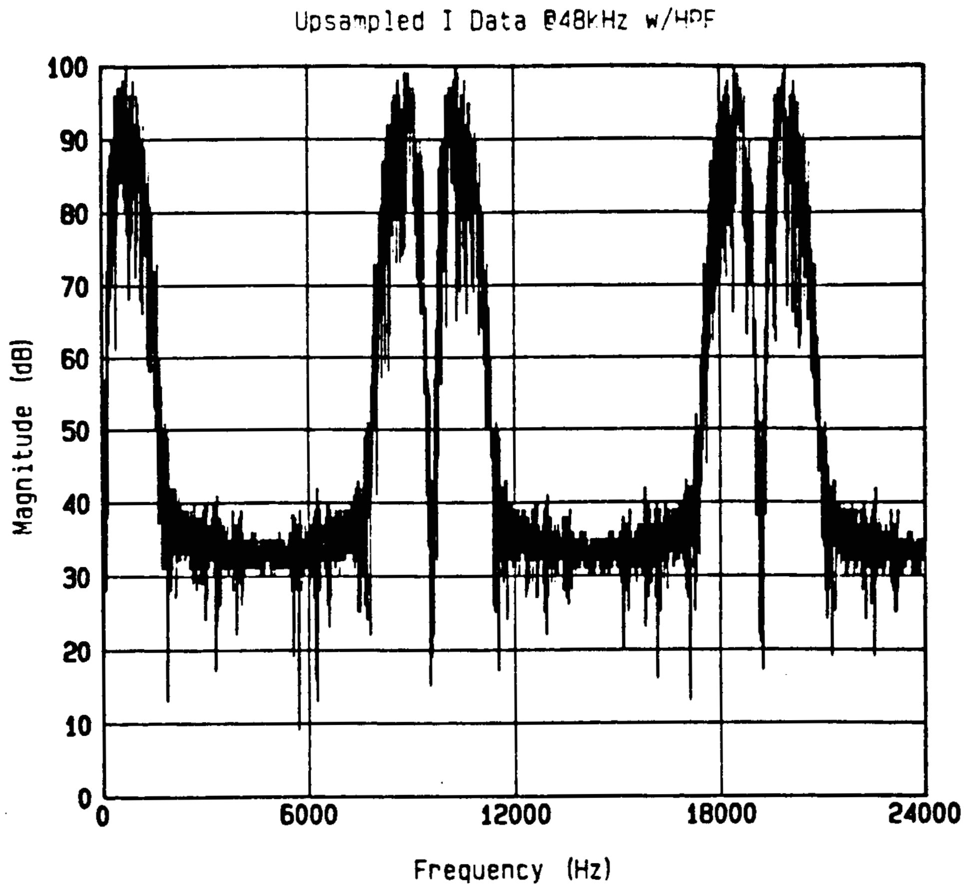


FIGURE 3.4(a) SPECTRUM OF 1:5 UPSAMPLED 9.6 kHz INPHASE SIGNAL

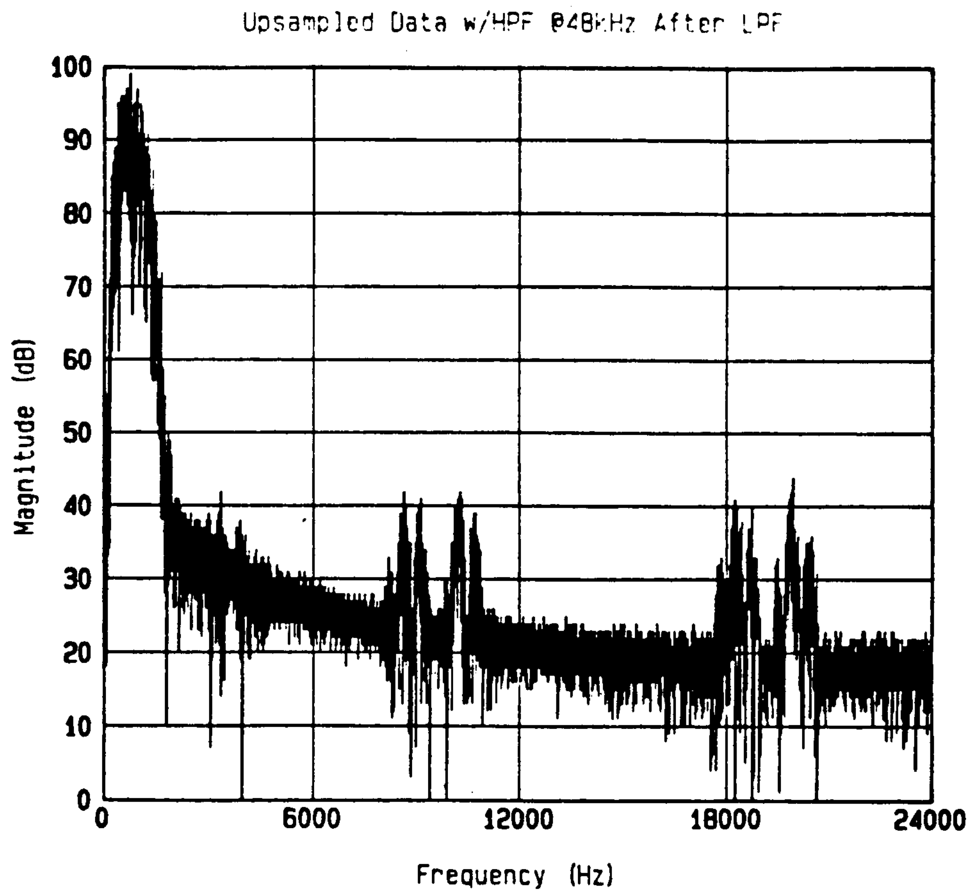


FIGURE 3.4(b) LOWPASS FILTERED SPECTRUM OF 1:5 UPSAMPLED 9.6 kHz INPHASE SIGNAL

3.1.2 Demodulator

The demodulator of Figure 3.5 was simulated in modular form with each block representing a different signal processing task that could be tested separately for functionality. A specific effort was made during simulation development to excercise system parameters for optimization of the real-time demodulator design. The efficiencies of multi-rate processing were emphasized.

Since the analog-to-digital converter is operating at a sampling frequency of 48 kHz, four times the IF frequency, all adjacent samples are in time quadrature. Translation to baseband is achieved by multiplying with a square wave a four sample period which, in the digital domain, is spectrally identical to a sinusoid of the same frequency. Every other sample of the baseband signal is then separated into in-phase and quadrature components sampled at a 24 kHz rate. Each component is divided into two streams. One undergoes pilot recovery and the other is delay equalized to match the lowpass filtering in the pilot processing.

The advantage of lowering the sampling frequencies clearly results in reduced digital processing cost. Decimation in the signal paths leading to the pilot recovery lowpass filters significantly eases both the processing burden and the digital filter design. A decimation ratio of 5:1 causes no aliasing and improves the lowpass cut-off to sampling frequency ratio, allowing a much lower order filter implementation. The recovered pilot streams are subtracted off from the delayed data and estimates of the channel phase perturbations are passed to the detection algorithm. The pilot processing and data detection are simulated essentially as described in section 2.1.2; however, since there are different processing rates in the demodulator, some method must be chosen to handle the different rate boundaries. A full interpolation to reconstruct the lower rate signal is expensive in processing cost. A zero-order hold is simple to implement but has a poor frequency characteristic as higher frequency images roll off with only a sin(x)/x response. As a result, a first-order hold is implemented providing much better attenuation of images. The processing cost is relatively low requiring only a one sample delay in the pilot path and a linear interpolation.

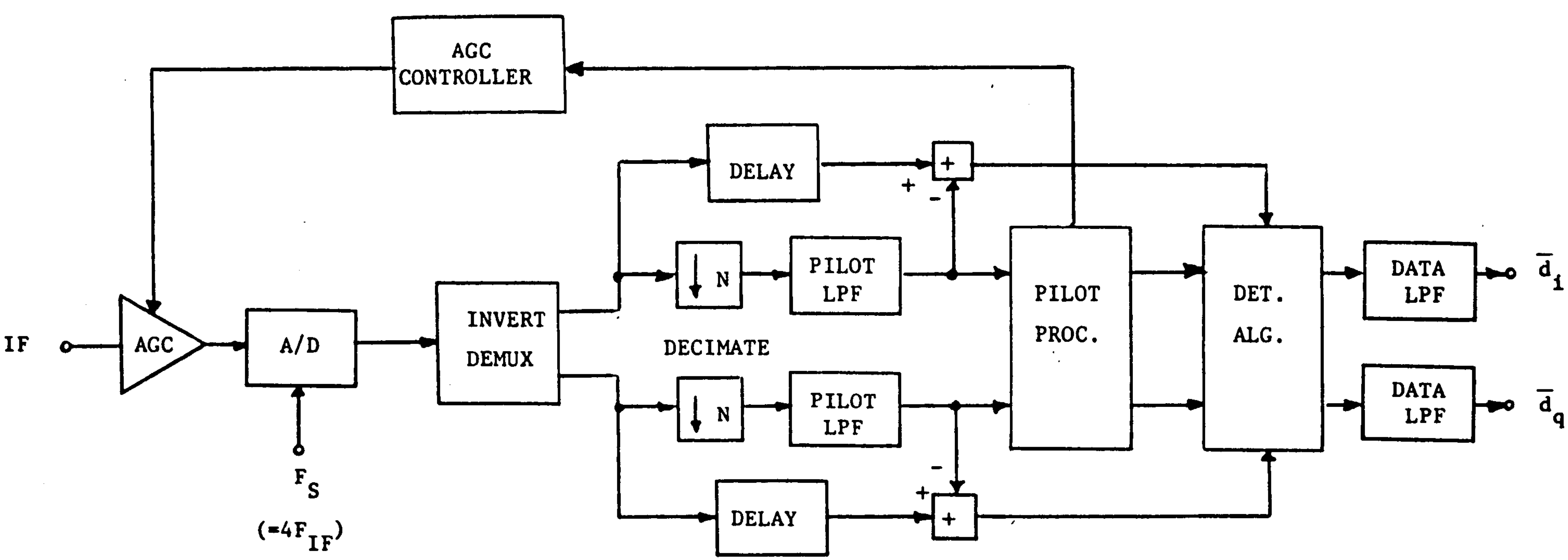


FIGURE 3.5 DIGITAL MTCT DEMODULATOR

The results of the simulated demodulator output are compared with respect to two performance criteria : recovered pilot quality and data eye quality. Two different pilot recovery filters were used with 3 dB cut-off frequencies representing maximum desired pilot bandwidth, 150 Hz, and minimum usable bandwidth, 80 Hz. The filter characteristics are summarized in table 3.1. The transmitted pilot level is 0.5 and the nominal eye opening is 2.0.

Figure 3.6(a) shows the recovered pilot using the 150 Hz lowpass filter without the benefit of a highpass filter on the transmit end. The large variation from the transmitted level is due solely to data modulation of the pilot. If the pilot of Figure 3.7(a) is compared to this one, a significant decrease in variance is observed. Although data energy continues to leak into the pilot recovery channel, the amount is greatly reduced at the transmit side by the Manchester encoding. The large reduction in pilot variance suggests that there is little data energy remaining in the frequency band below 80 Hz.

Figures 3.6(b) and 3.7(b) show the eye diagrams corresponding to the recovered pilots. Both show the ISI introduced by the demodulator at the detection algorithm. The eye pattern generated from the system employing the 150 Hz recovery filter is particularly bad due to the large data modulation induced variance.

With the addition of a transmit highpass filter, the demodulator recovered pilots more closely approach the transmitted levels. Figure 3.8(a) shows the pilot obtained with the 150 Hz recovery filter and Figure 3.9(a) shows the pilot recovered through the 80 Hz lowpass filter. Both pilots exhibit considerable improvement due to the inclusion of the highpass filter.

The respective data eye patterns are shown in Figures 3.8(b) and 3.9(b). A comparison between the detected eye diagram obtained with the 80 Hz recovery filter and the transmitted data eyes of Figures 3.2(a) and 3.3(a) indicates negligible distortion introduced by the demodulator.

A configuration which was not simulated but potentially results in reduced ISI at the demodulator consists of transmit highpass filtering at a frequency of 80 Hz and also employing an 80 Hz pilot recovery filter. This

TABLE 3.1

PILOT RECOVERY FILTER CHARACTERISTICS

150 Hz Pilot Recovery Filter:

3 dB bandwidth	150 Hz
40 dB bandwidth	300 Hz
passband ripple	0.1 dB
sampling frequency	4.8 kHz

80 Hz Pilot Recovery Filter:

3 dB bandwidth	80 Hz
40 dB bandwidth	160 Hz
passband ripple	0.1 dB
sampling frequency	4.8 kHz

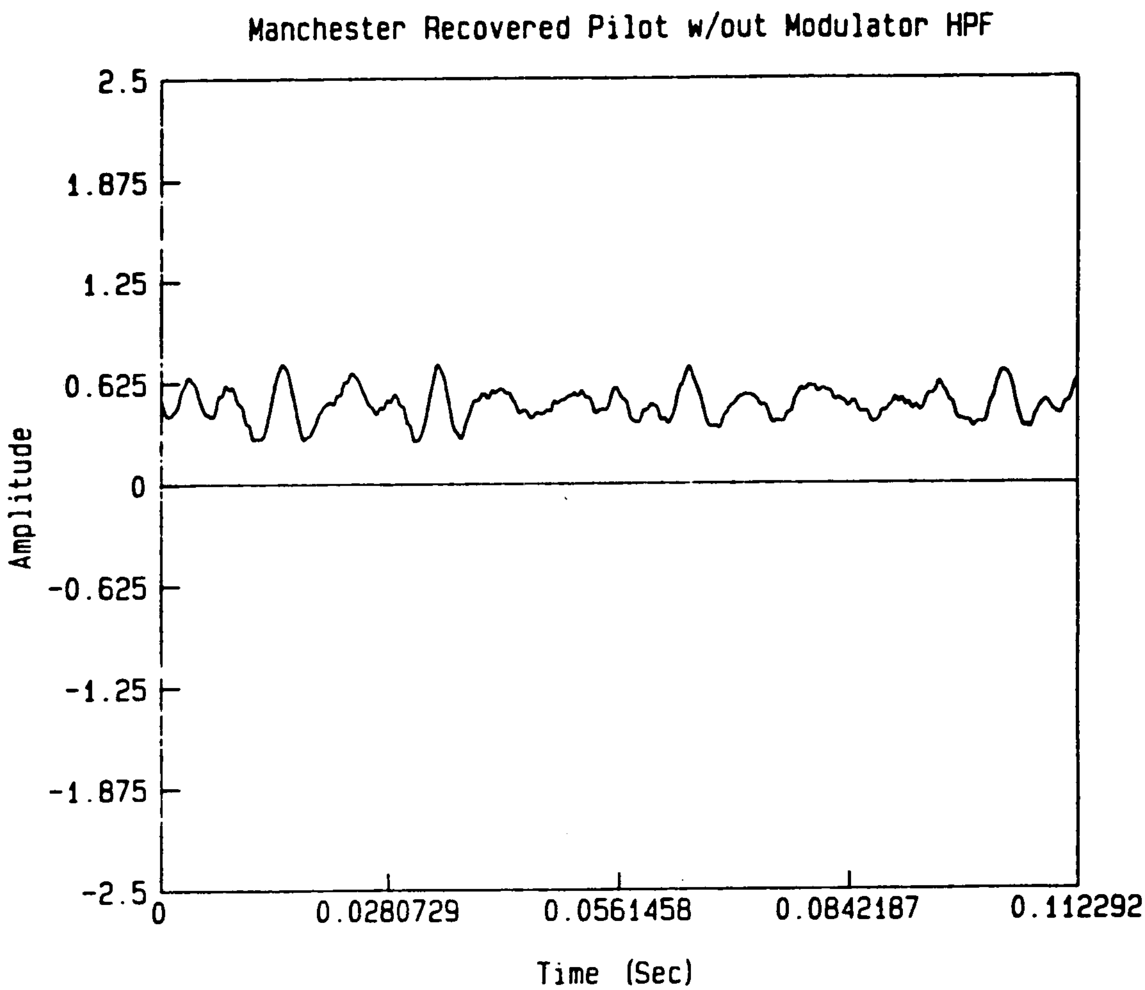


FIGURE 3.6(a) MTCT RECOVERED PILOT, 150 HZ PILOT LOWPASS FILTER, NO TRANSMIT HIGHPASS FILTER

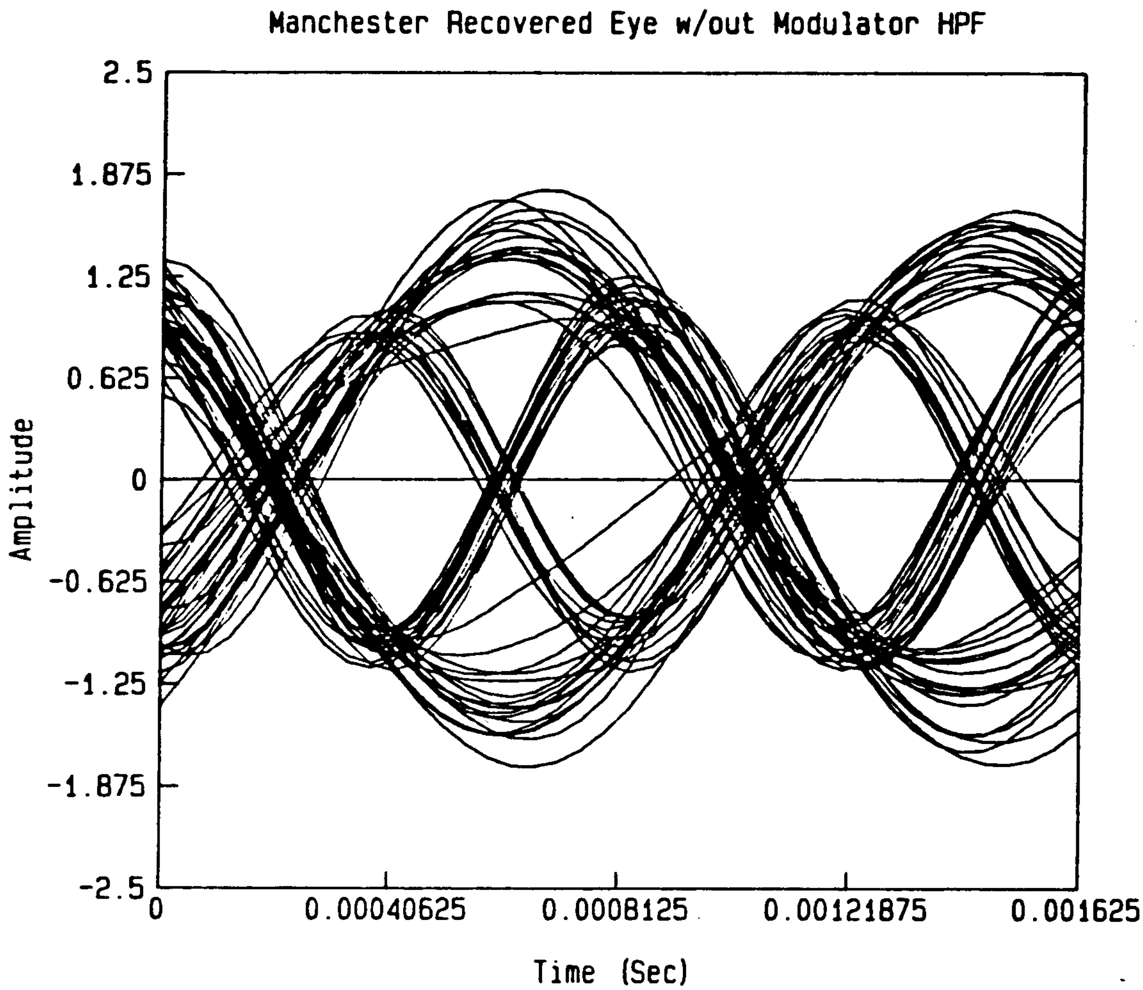


FIGURE 3.6(b) MTCT RECOVERED EYE FOR THE PILOT OF FIGURE 3.6 (a)

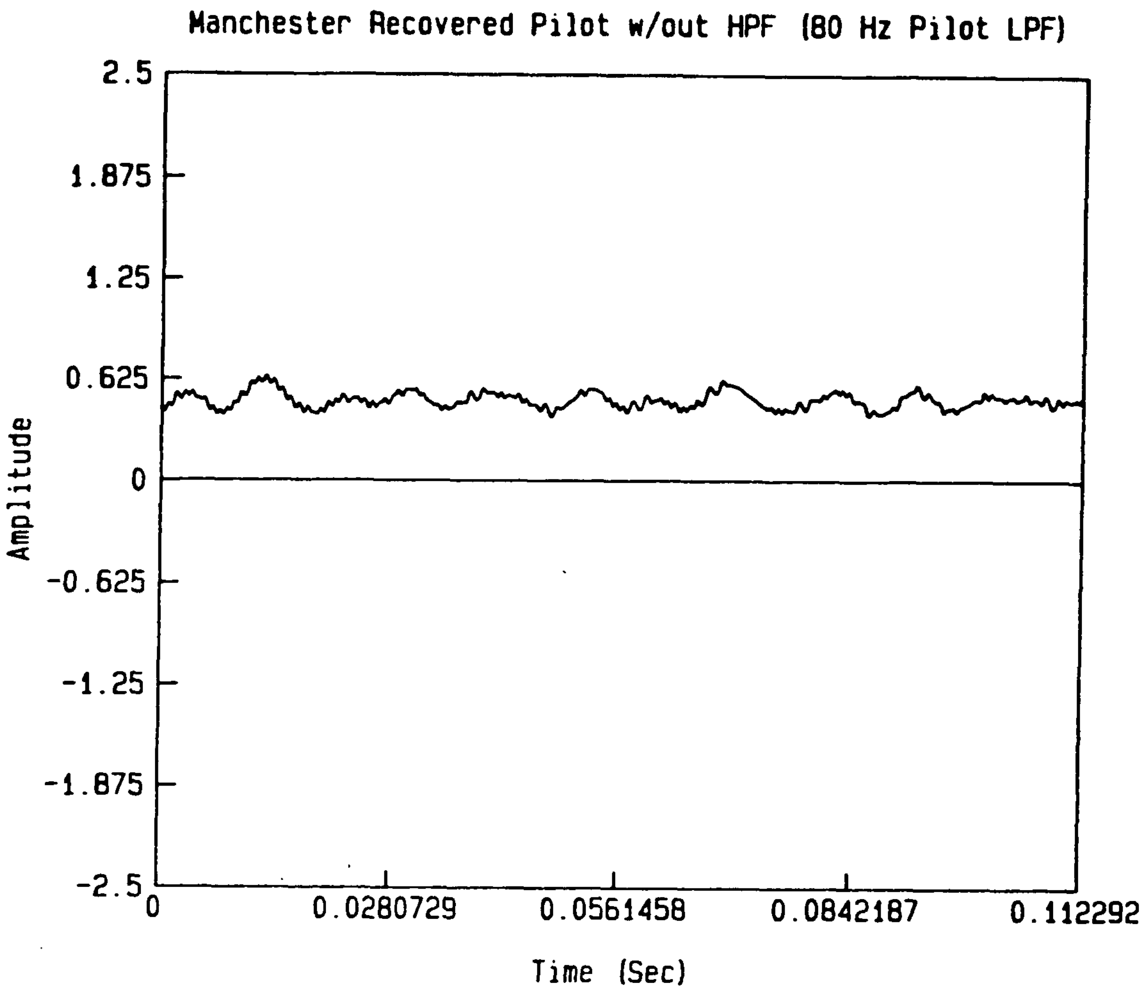


FIGURE 3.7(a) MTCT RECOVERED PILOT, 80 HZ PILOT LOWPASS FILTER, NO TRANSMIT HIGHPASS FILTER

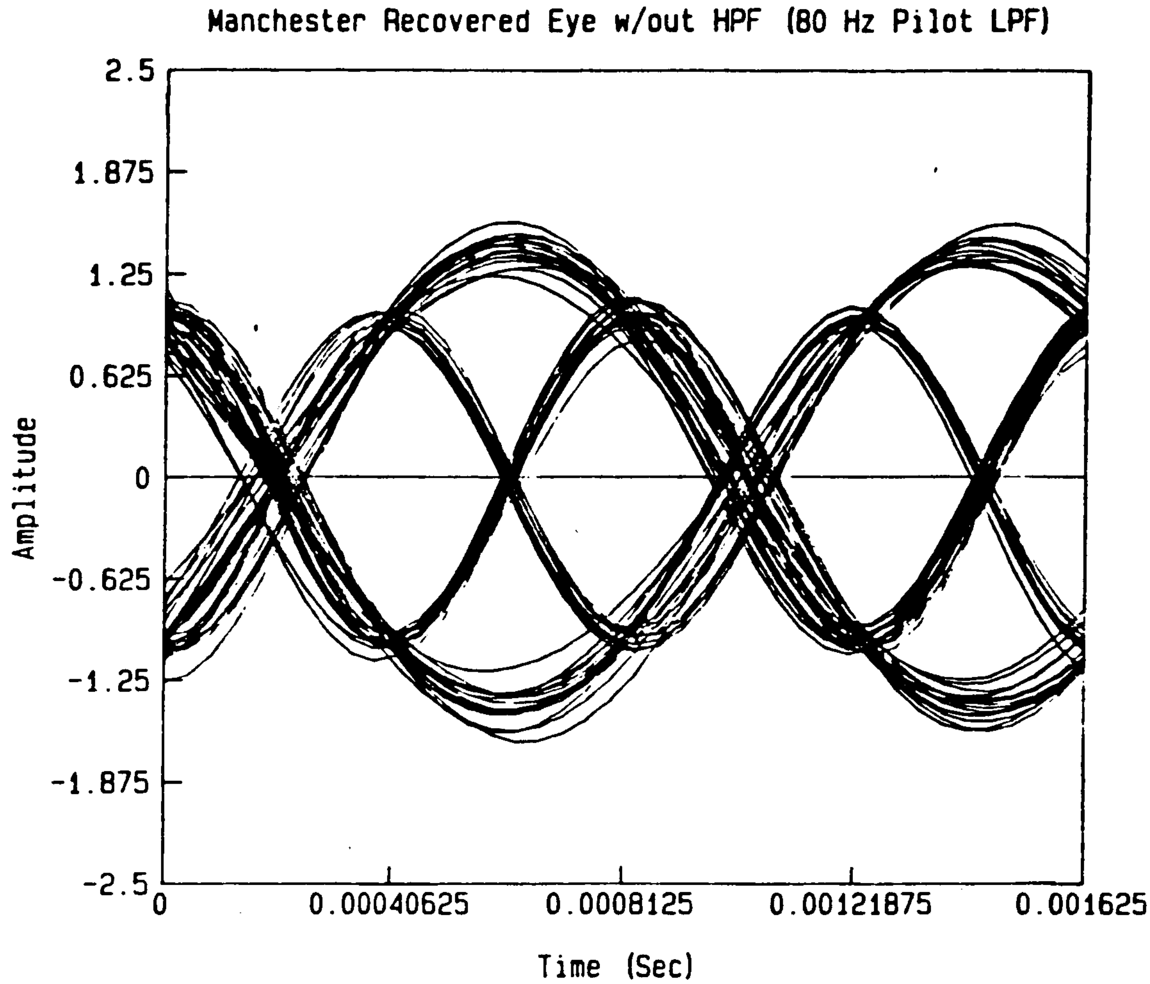


FIGURE 3.7(b) MTCT RECOVERED EYE FOR THE PILOT OF FIGURE 3.7(a)

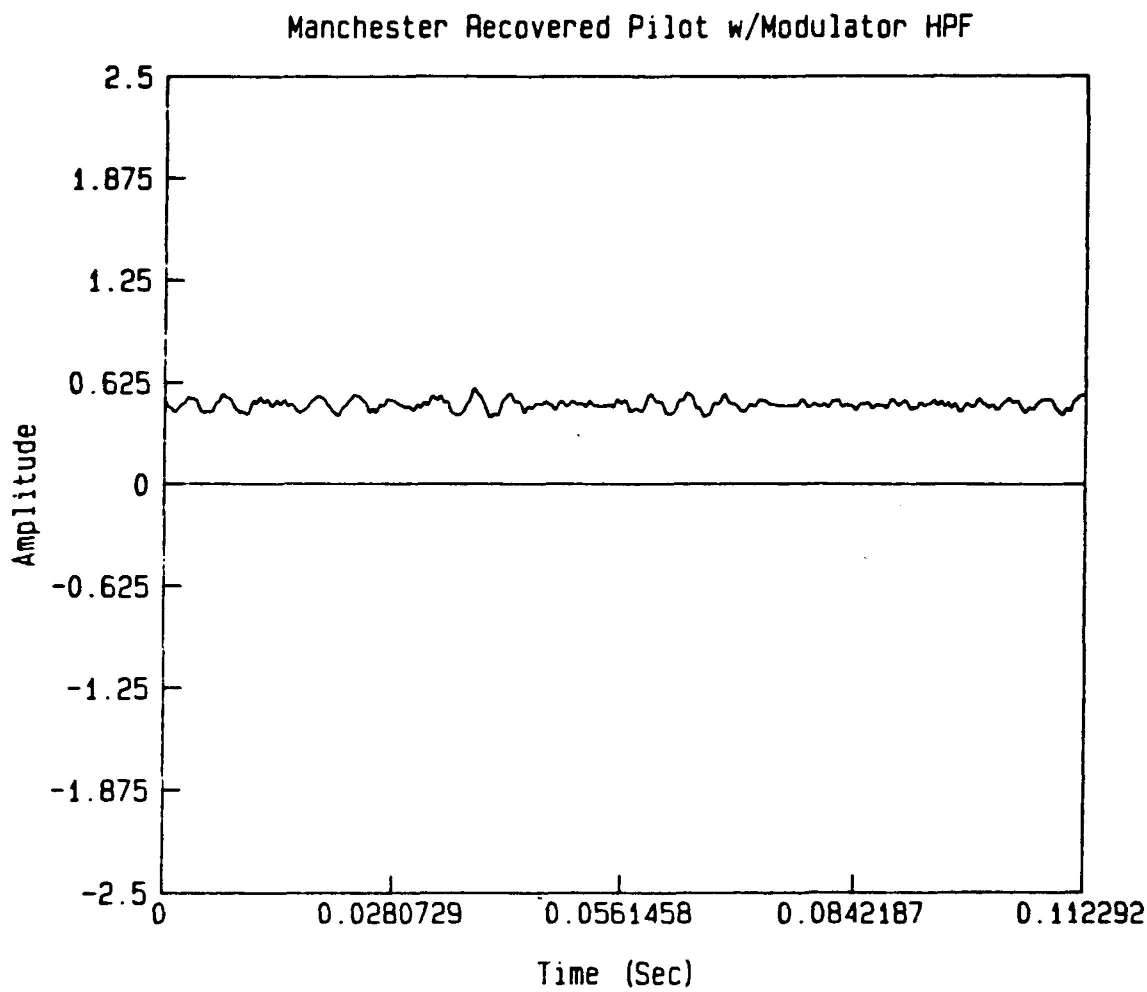


FIGURE 3.8(a) MTCT RECOVERED PILOT, 150 HZ LOWPASS FILTER AND TRANSMIT HIGHPASS FILTER

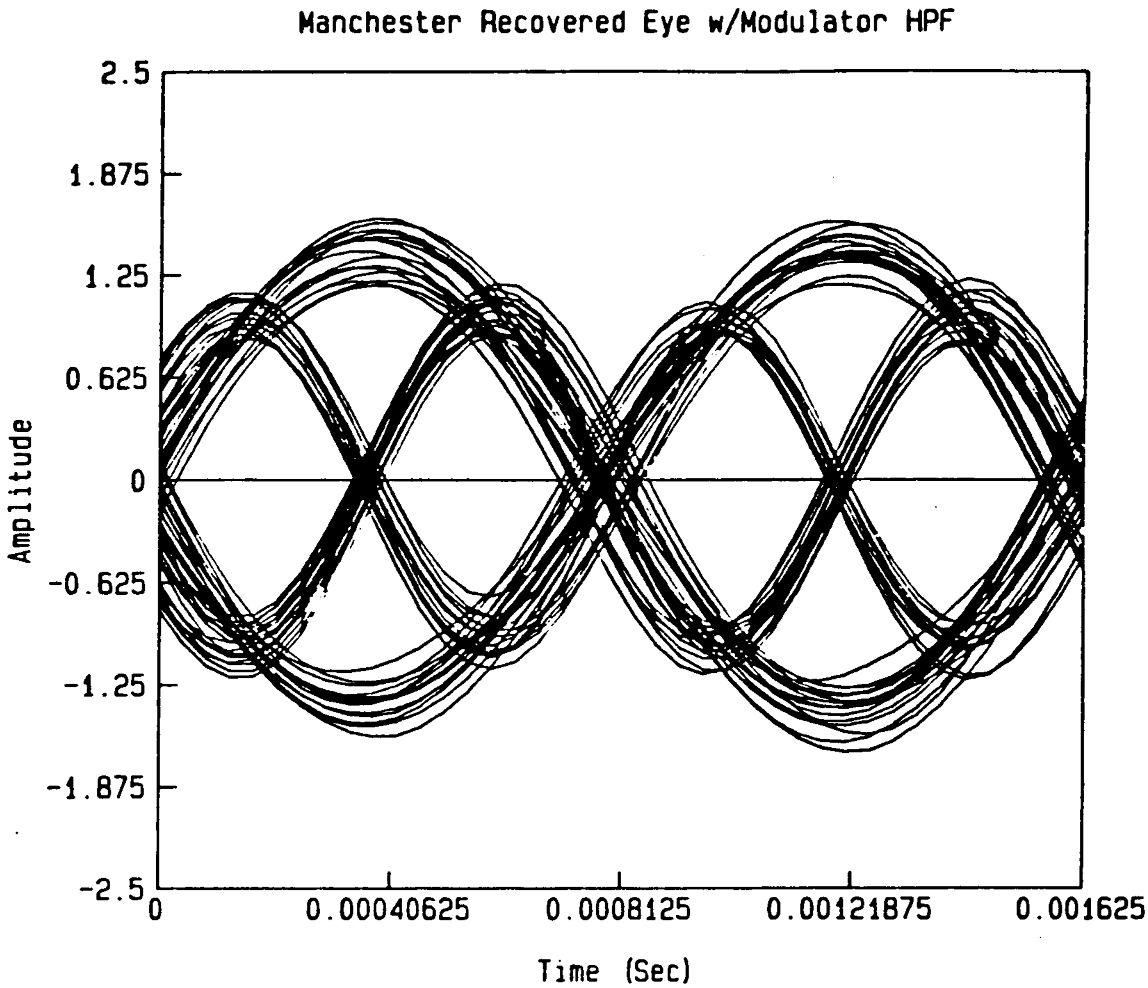


FIGURE 3.8(b) MTCT RECOVERED EYE FOR THE PILOT OF FIGURE 3.8(a)

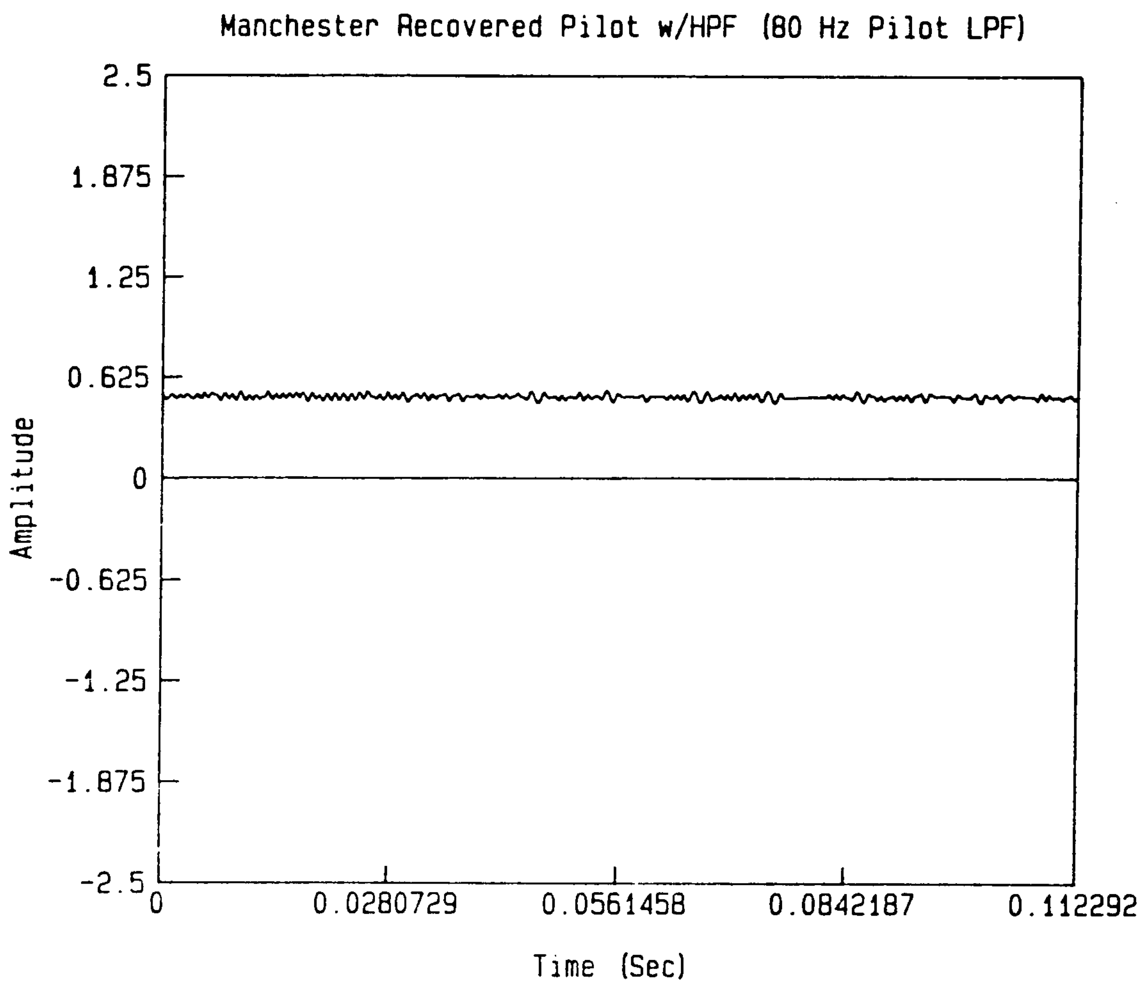


FIGURE 3.9(a) MTCT RECOVERED PILOT, 80 HZ LOWPASS FILTER AND HIGHPASS TRANSMIT FILTER

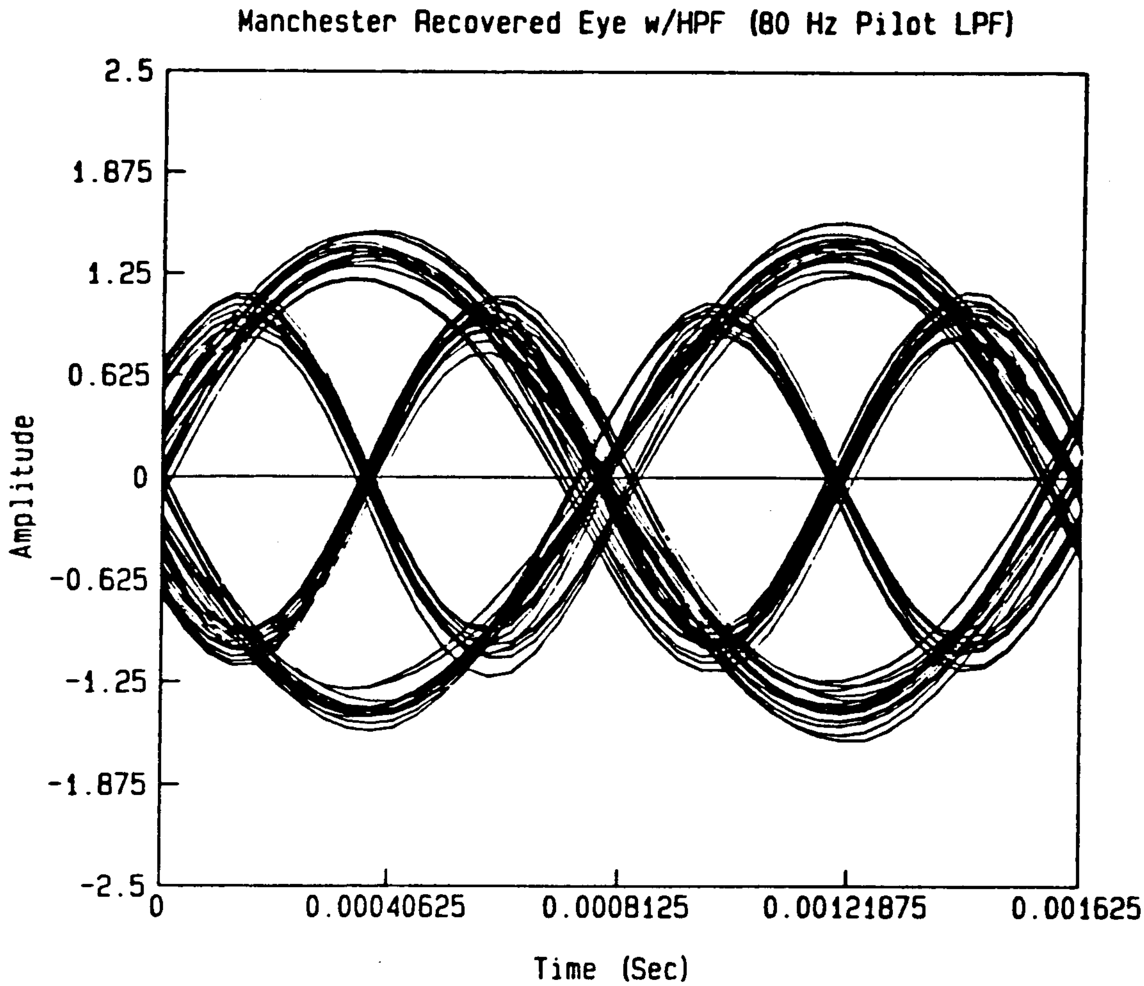


FIGURE 3.9(b) MTCT RECOVERED EYE FOR THE PILOT OF FIGURE 3.9(a)

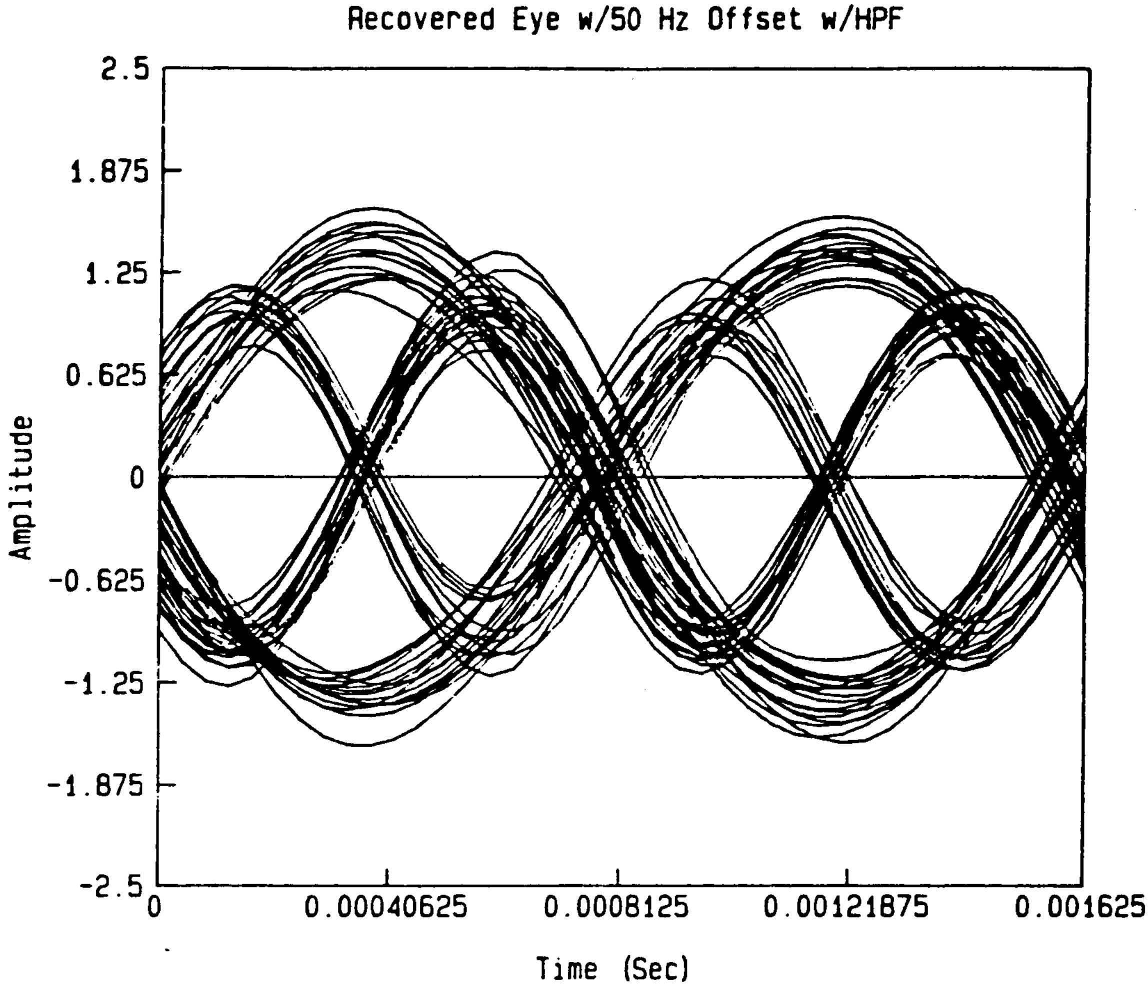


FIGURE 3.10(a) MTCT RECOVERED EYE FOR A 50 HZ FREQUENCY OFFSET AND A 150 HZ PILOT LOWPASS FILTER

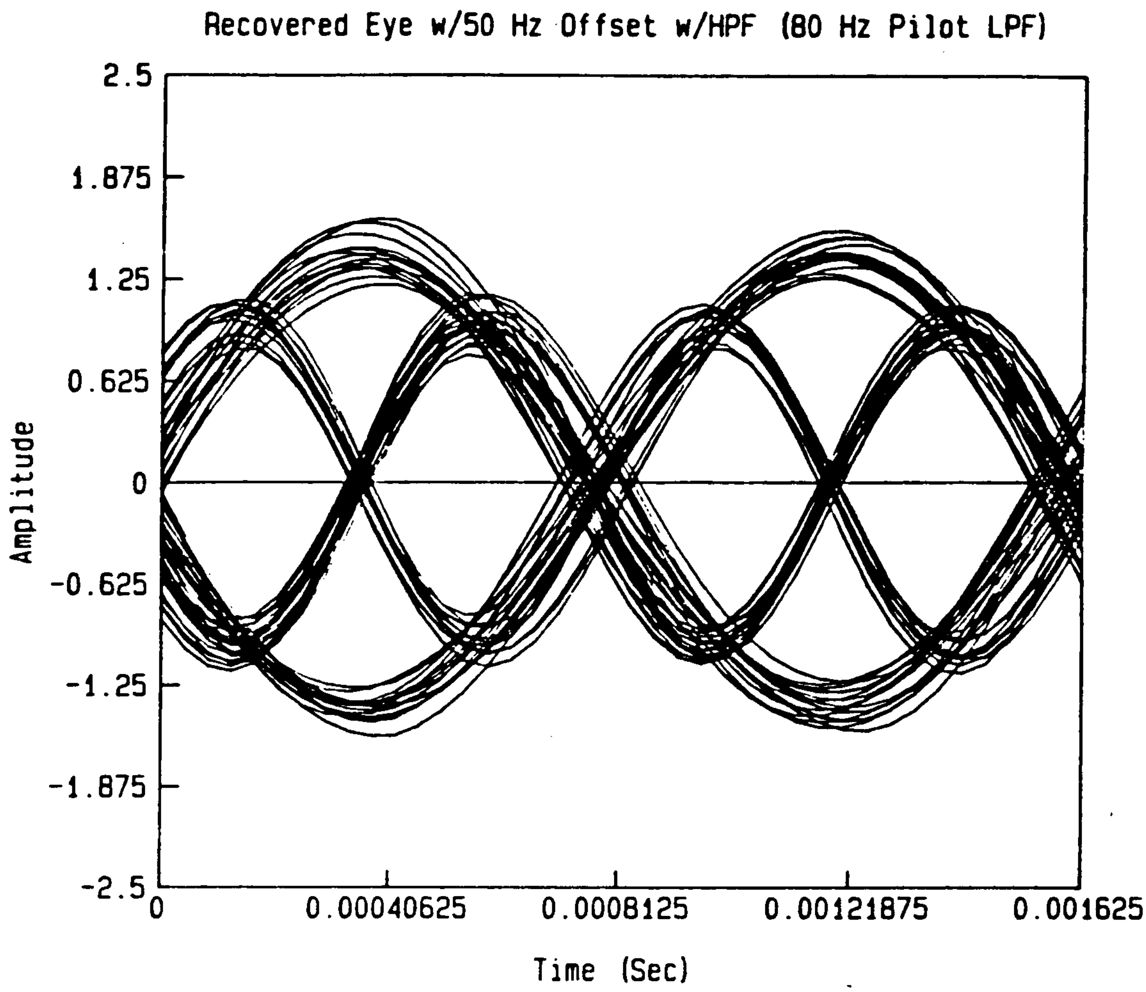


FIGURE 3.10(b) MTCT RECOVERED EYE FOR A 50 HZ FREQUENCY OFFSET AND AN 80 HZ LOWPASS FILTER

would result in a recovered pilot with slightly more variance than the one of Figure 3.9(a); however, the data eye should display less ISI since a significant part of the data spectrum (80-150 Hz) is left intact.

The demodulator performance was also examined after introducing a 50 Hz frequency offset to the modulated signal. Figures 3.10(a-b) show the detected data eyes obtained with the 150 Hz and 80 Hz lowpass filters respectively. The performance exhibited by the MTCT configuration indicates the ability to correct for a considerable frequency displacement.

3.2 Subcarrier TCT

3.2.1 Modulator

As an alternative to the MTCT transmitter, the STCT modulator of Figure 2.4 was fully simulated. The STCT modulator relies on the subcarrier modulation to create the spectral null for pilot insertion; therefore, its design is considerably simplified by omitting both the Manchester encoding and the highpass filters. Frequency domain raised-cosine pulse shaping is again used in the STCT system. An excess bandwidth fraction of 0.4 is employed for pulse shaping and a 960 Hz subcarrier, for QPSK modulation in order to meet the single-sided spectral occupancy requirement of a 40 dB attenuation at 1.8 kHz from the center frequency. The modulator output is a 48 kHz sampled stream translated to a 12 kHz IF.

The baseband data eye is shown in Figure 3.11(a) and 3.11(b) shows the corresponding spectrum. Figure 3.12 shows both the QPSK modulated data onto the subcarrier and the spectral null. Since no energy has been removed from the data signal, unlike the MTCT system, the transmitted data is free of ISI.

3.2.2 Demodulator

The translation process from IF to baseband in the STCT demodulator was changed slightly from the one used in the MTCT system. Demodulation was achieved without a local reference by taking advantage of the fact that the data bandwidth is small in comparison to the sampling frequency. In the STCT

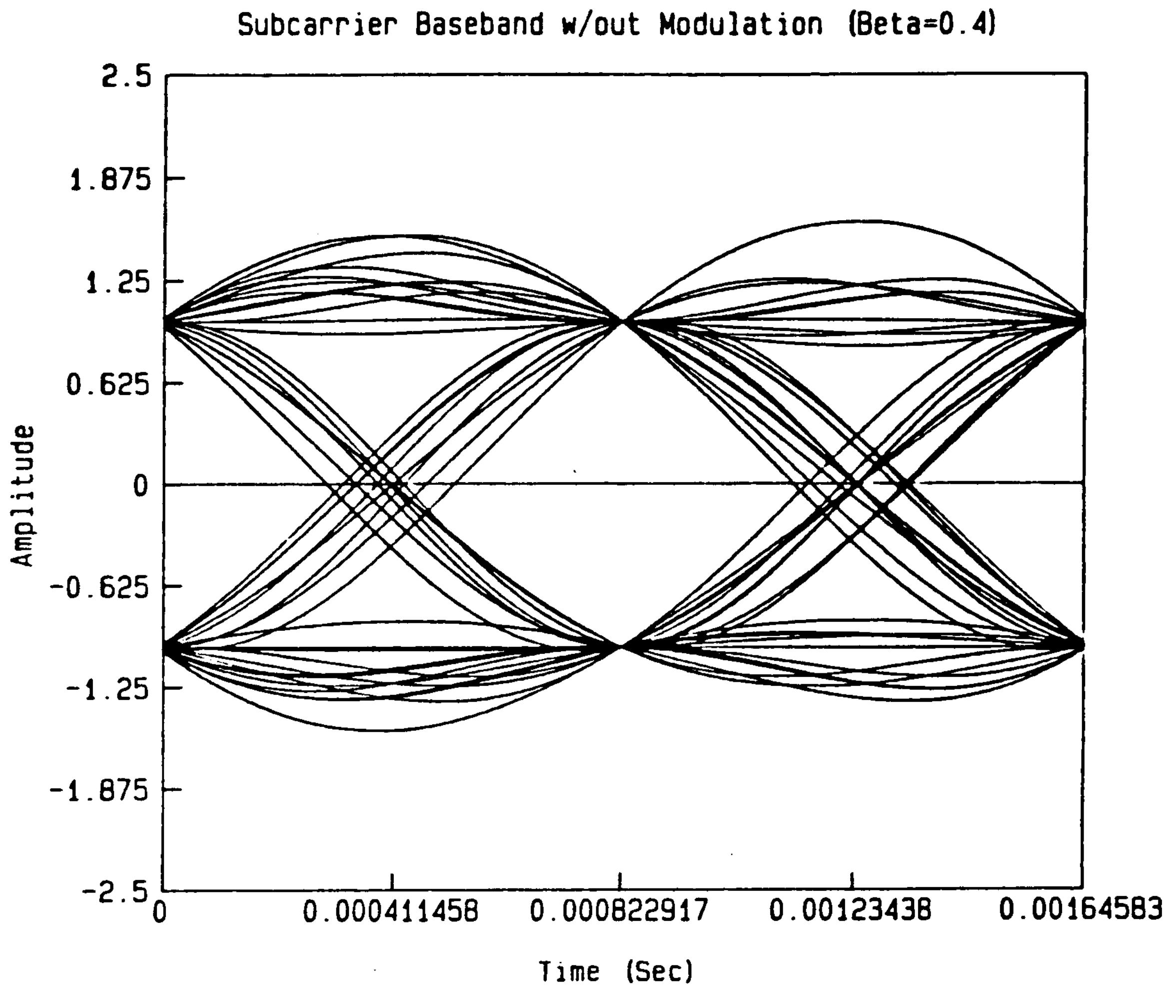


FIGURE 3.11(a) STCT TRANSMIT DATA EYE

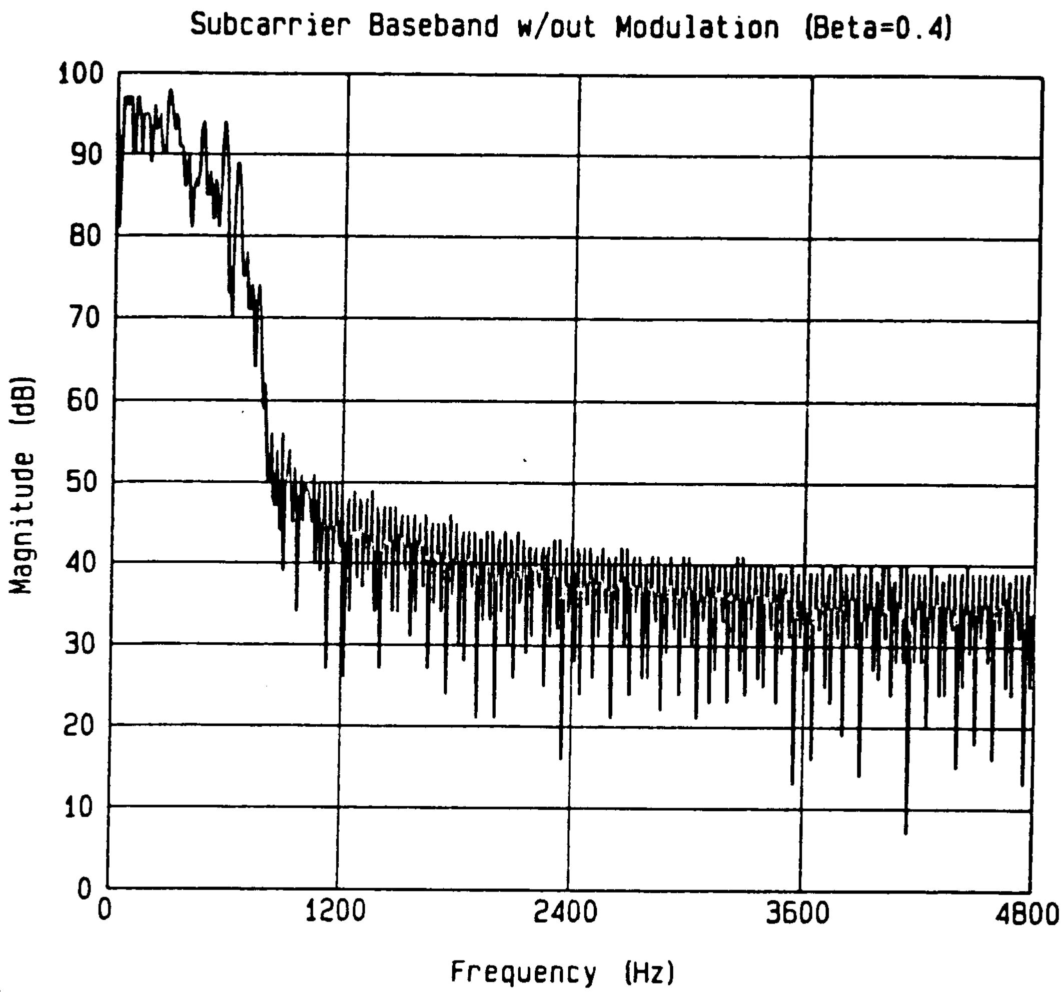


FIGURE 3.11(b) FREQUENCY SPECTRUM OF FIGURE 3.11(a)

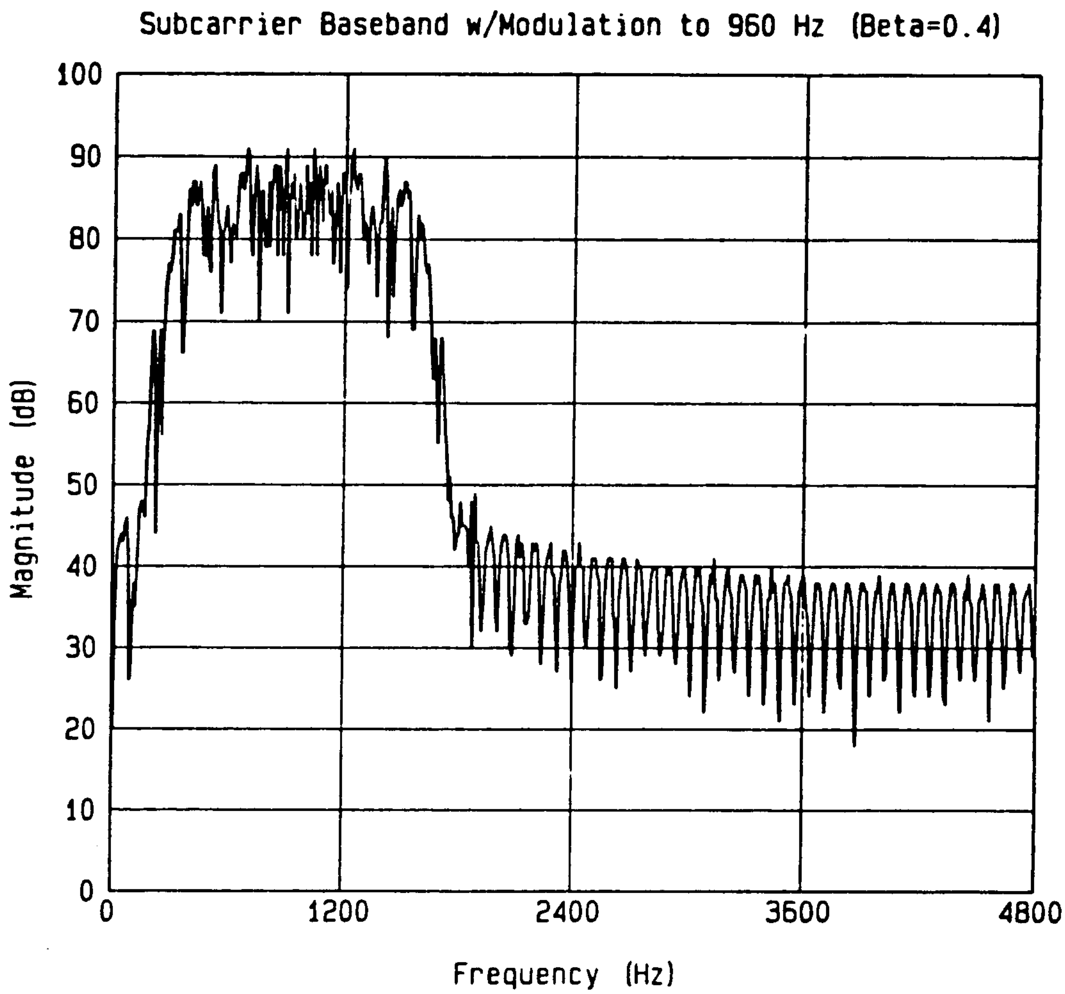


FIGURE 3.12 SUB-CARRIER QPSK FREQUENCY SPECTRUM

simulation, a ratio of 4:1 was employed and adjacent IF input sample pairs were translated to 12 kHz in-phase and quadrature baseband components. Figures 3.13(a-c) detail the demodulation process. A further decimation of 5:1 in the signal paths leading to the pilot recovery lowpass filters was used causing aliasing in the data but not in the pilot. The aliased portion of the spectrum is located in the stopband of the recovery filter and contributes no negative effects, see Figures 3.14(a-b). The final processing rates in the STCT simulation were 12 kHz for the data arms and 2.4 kHz for the pilot channels (these rates can be applied to the MTCT simulation as well).

The results of the subcarrier demodulator simulation are summarized in Figures 3.15(a-b) and 3.16. Figure 3.15(a) shows a recovered pilot with virtually no data modulation corrupting it, resulting in the corresponding recovered eye pattern of Figure 3.15(b) which shows no ISI. A frequency offset test was also performed with the subcarrier simulation. Figure 3.16 shows the recovered data eye which displays noticeable distortion. Some preliminary investigation seemed to indicate that ISI is produced by imperfect cancellation of the frequency offset double term product in the detection algorithm. The distortion introduced by this term is assumed also to be present in the MTCT demodulator.

Table 3.2 compares the recovered pilot variances for both the MTCT and STCT systems. All variances are referenced to the recovered subcarrier pilot variance level.

4. MODEM HARDWARE IMPLEMENTATION

4.1 Manchester Encoded TCT

The Manchester encoded TCT was the method chosen by JPL to be followed through to a hardware realization. Consequently, the MTCT modem is the major topic of this section. To date a stand-alone board that serves as the digital modulator for both the MTCT and STCT systems has been completed and is fully tested and functional. The required RF circuitry for the modulator and demodulator has also been designed and tested. The transmitter RF portion has

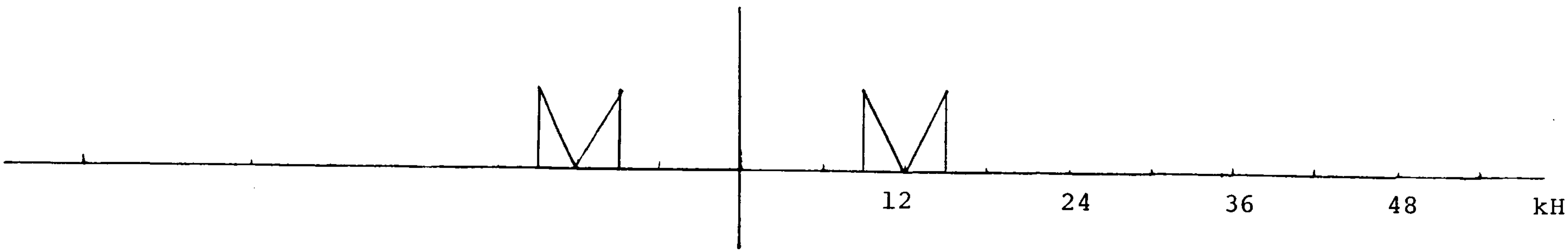


Figure 3.13(a) Analog IF Spectrum

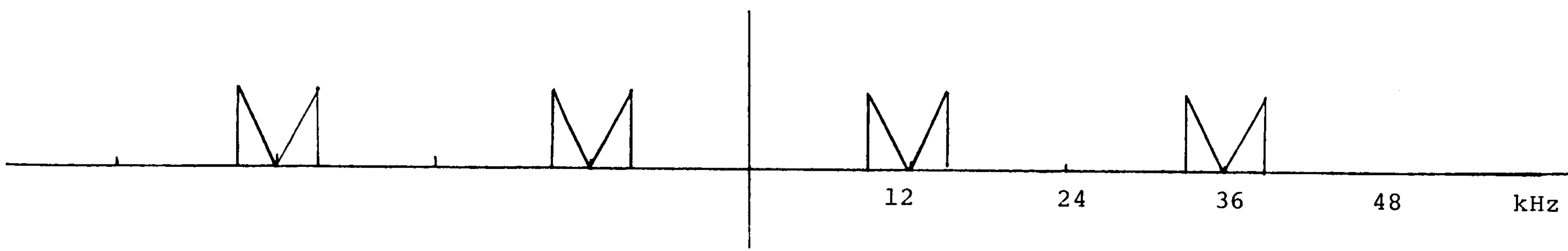


Figure 3.13(b) Spectrum Sampled at 48 kHz

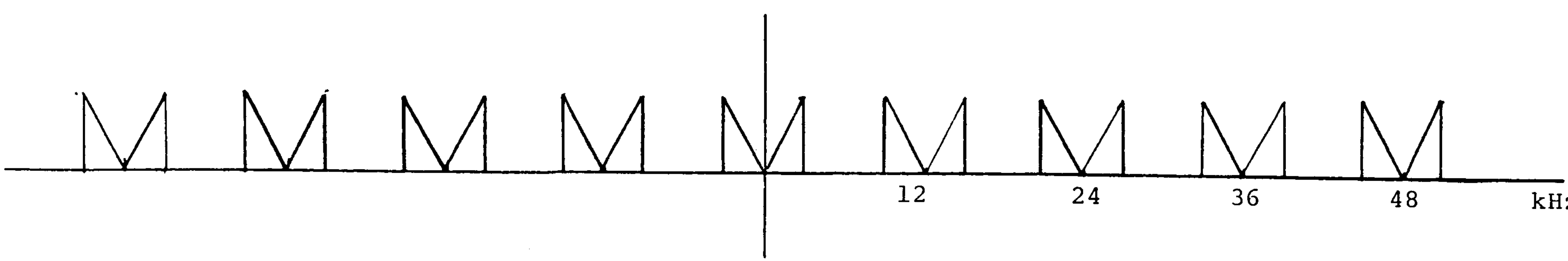


Figure 3.13(c) Demodulated Spectrum Through 4:1 Decimation

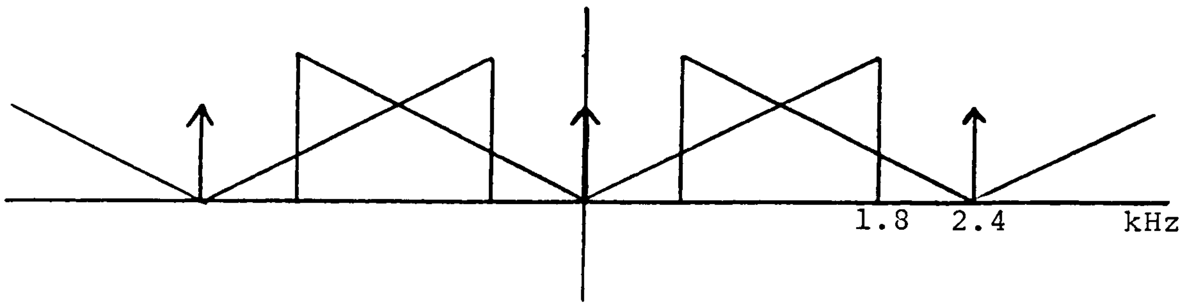


Figure 3.14(a) Aliased Baseband Spectrum

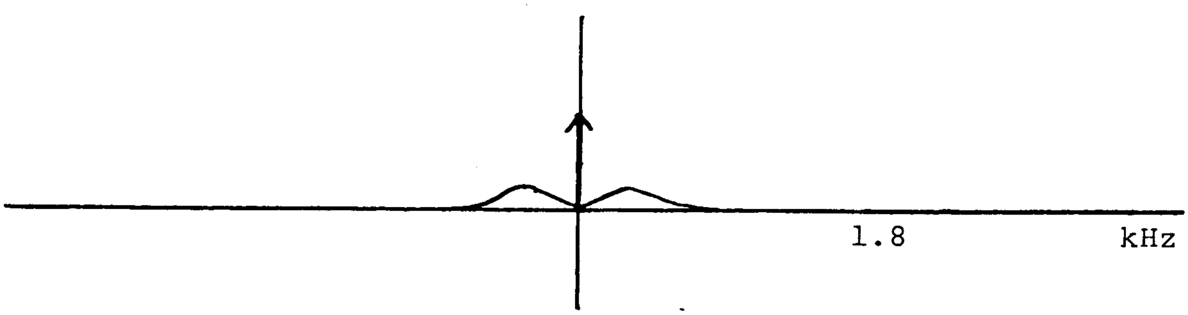


Figure 3.14(b) Recovered Pilot After Lowpass Filtering

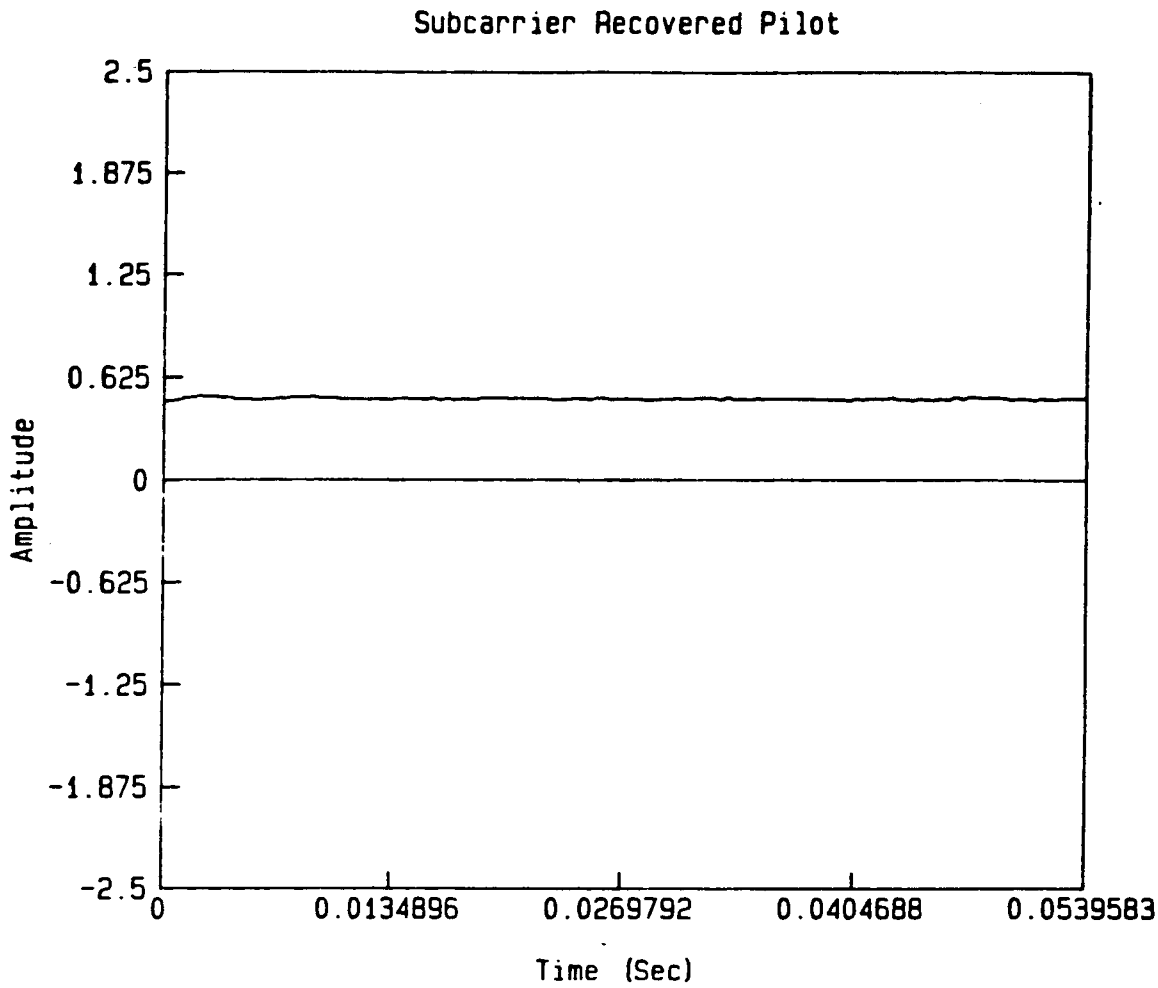


FIGURE 3.15(a) STCT RECOVERED PILOT, 150 HZ PILOT LOWPASS FILTER

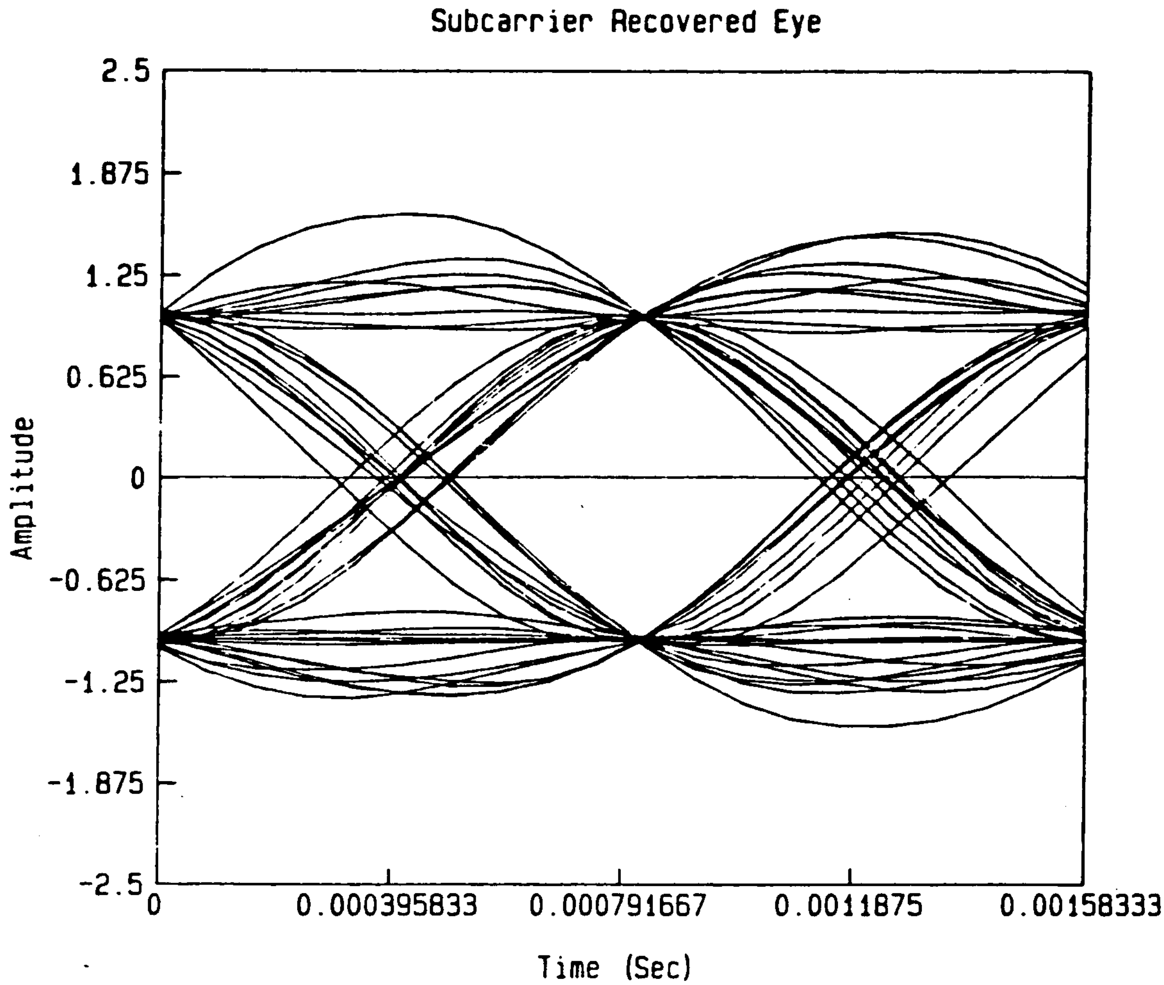


FIGURE 3.15 STCT RECOVERED EYE FOR THE PILOT OF FIGURE 3.15(b)

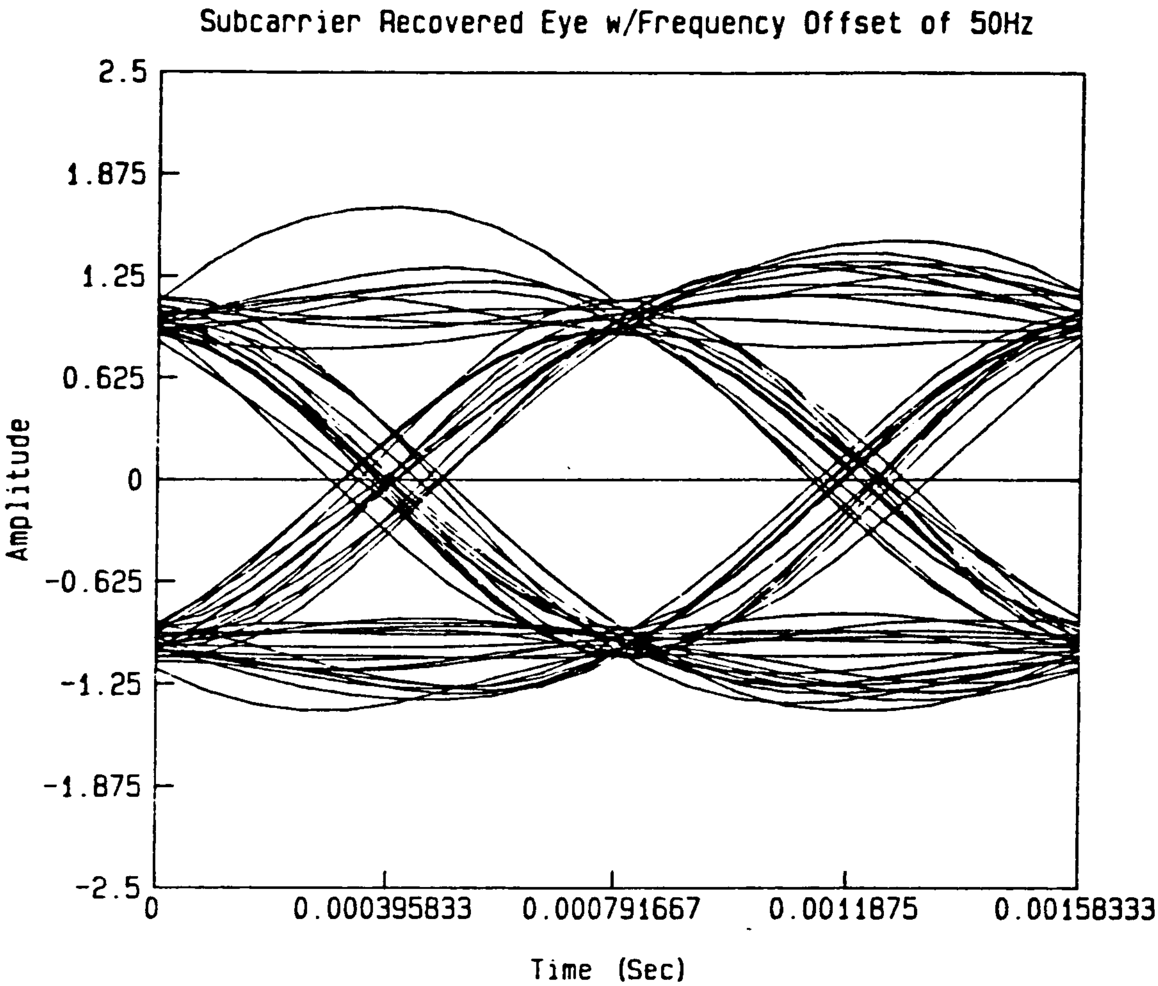


FIGURE 3.16 STCT RECOVERED EYE FOR A 50 HZ FREQUENCY OFFSET

MANCHESTER TCT

TX HPF	PILOT REC. LPF	RELATIVE PILOT LEVEL
NO	150 HZ	22.8 DB
YES	150 HZ	17.5 DB
NO	80 HZ	16.1 DB
YES	80 HZ	6.2 DB

SUB-CARRIER TCT

TX HPF	PILOT REC. LPF	RELATIVE PILOT LEVEL
NO	150 HZ	0.0 DB

Table 3.2 Recovered Pilot Variances in dBs

been integrated with the stand-alone modulator board and and is operative, as will be shown later. A circuit board design had been initiated for the MTCT demodulator and the TMS320 code is written, however, due to insufficient time to complete construction within the project schedule, it was decided by JPL not to follow it through to completion. The MTCT TMS320 software used in the premodulation processing is included in Appendix II.

Since the hardware implementation of the modulator was discussed previously in both the first and second interim reports, only the salient features will be presented in the following section.

4.1.1 MTCT Modulator Implementation

A block diagram of the final stand-alone digital modulator is shown in Figure 4.1. All processing tasks are performed by the Texas Instruments TMS320, the remaining components of the board are necessary for the proper function of the TMS320. The program code is held in the 2K of EPROM and is read into the 2K of RAM upon the booting of the system. In this way, a slow access time, UV erasable PROM can be used in conjunction with fast RAM to avoid usage of once-only programmable bipolar PROM. The I/O block consists of three latches which control the two system outputs and single system input. The two remaining blocks pertain to the control timing of the processing. The Master Timing block includes a 20 MHz clock for the TMS32010 processor chip, while the other I/O timer strobes the TMS320 at a rate of 9.6 kHz., indicating that it is time for the system to release an inphase and quadrature sample pair to the QPSK modulator. Also, on every fourth set of outputs, a new data bit is read in.

The details of the modulator code were presented in the First Interim report [1] and will only be briefly summarized here. The TMS320 accepts a binary input at a rate of 2.4 kbps. These input bits are split into even and odd data streams prior to Manchester encoding. Encoding is performed only after two data bits have been input which has the effect of synchronizing the inphase and quadrature streams. The next step in the modulation process is to shape these encoded bits.

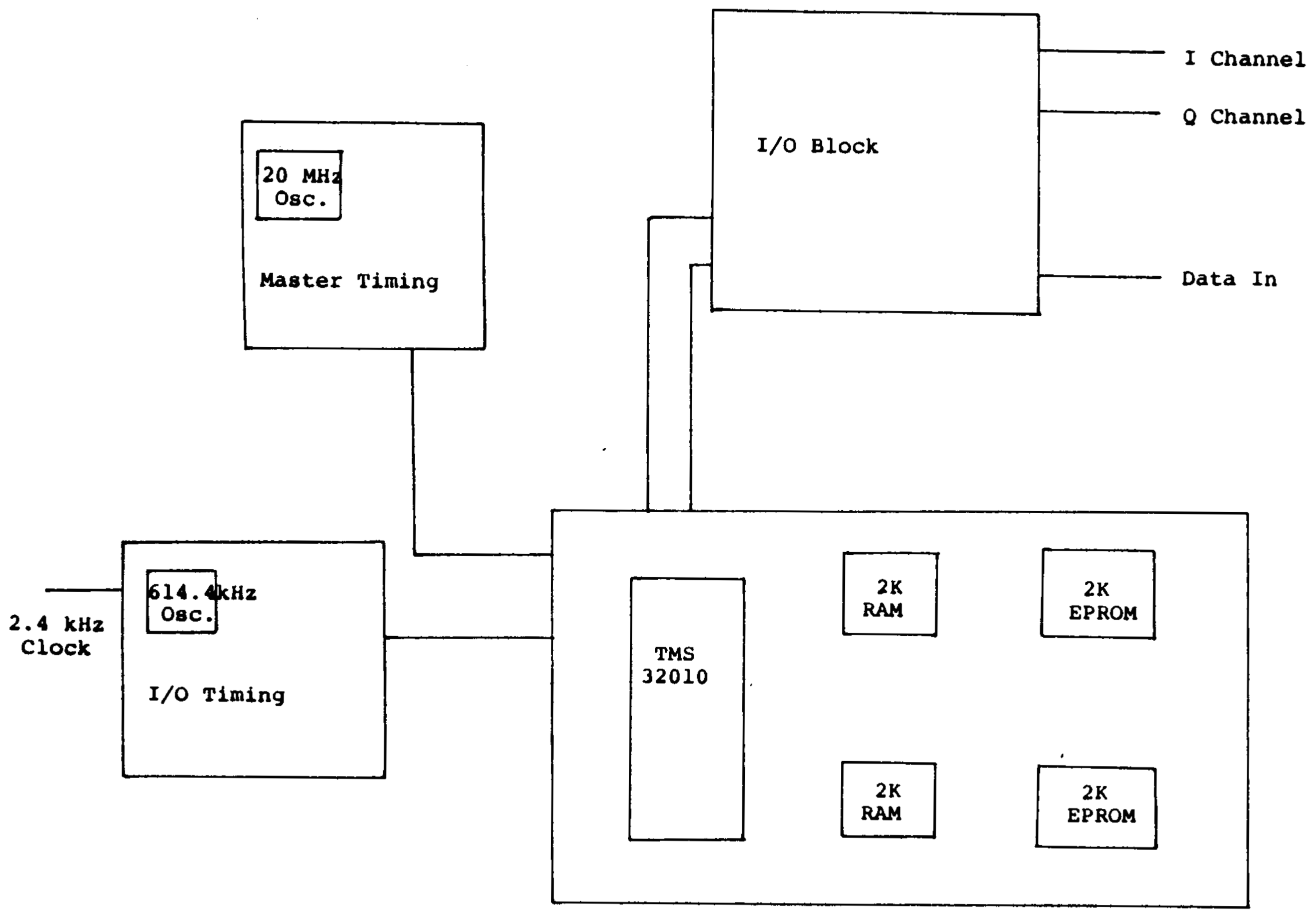


FIGURE 4.1 MTCT MODULATOR BOARD CONFIGURATION

The pulse-shaping chosen for implementation is the raised-cosine shape with an excess bandwidth fraction, β, of 0.5, as discussed in section 2.1. The raised-cosine pulse-shape is truncated such that it spans eight Manchester encoded data bits and is represented digitally by four evenly spaced samples per bit, for a total of thirty-two samples per pulse waveform. As will be shown presently, this representation of the raised-cosine pulse is sufficient to produce a transmit data 'eye' of the desired quality and spectral occupancy.

These thirty-two pulse-shape coefficients are stored in ROM and listed in Table 4.1. This table includes both the actual values of the coefficients as well as the scaled values used in the TMS320 implementation. The encoded data bits are pulse-shaped by simply multiplying these coefficients by the code bit in question. Therefore, a +1 is represented by the thirty-two coefficients that appear in Table 4.1, a -1 by their inverse. At any point in time, the output of the pulse-shaping section is simply the sum of the samples from all waveshapes that are non-zero at that instant in time. Due to the truncation of the pulse-shape, only the waveforms representing the eight most recent Manchester bits are non-zero and, hence, taken into consideration. Since there are four samples of the pulse-shape per Manchester bit period, there will be four outputs per I/Q stream from the processor chip per Manchester bit. The inphase and quadrature shaping is identical and operates independently on the separated even and odd data bit streams. Hence the I and Q shaping can be represented by the general equation:

$$s(t_i + j) = \sum_{n=t_i-7}^{t_i} cb_n * P(4(n-t_i + 8) + j) \qquad j = 1,2,3,4 \tag{4.1}$$

where t_i is the time index of the most recent Manchester bit, cb_n refers to the encoded data bits (even or odd), the index j is an output pointer which indicates which of the four outputs for this particular set of code bits is under consideration, and the P(.) terms are the raised-cosine pulse-shape samples listed in table 4.1. After all four output samples have been generated, a new code bit enters into play, and the oldest bit of the last nine is

Table 4.1

Pulse Shaping Coefficients

	Actual Value	Scaled Value
P(1) = P(32)	.00219	36
P(2) = P(31)	.00555	91
P(3) = P(30)	.00464	76
P(4) = P(29)	.000867	14
P(5) = P(28)	.00114	19
P(6) = P(27)	.01056	173
P(7) = P(26)	.0221	363
P(8) = P(25)	.01599	262
P(9) = P(24)	-.02533	-415
P(10) = P(23)	-.09172	-1503
P(11) = P(22)	-.1334	-2186
P(12) = P(21)	-.07953	-1303
P(13) = P(20)	.1159	1899
P(14) = P(19)	.4289	7028
P(15) = P(18)	.7587	12431
P(16) = P(17)	.9709	15908

discarded. Figure 4.2(a) shows the resulting eye diagram after Manchester encoding and raised-cosine pulse shaping. Note that at the sampling instants there is no intersymbol interference. Figure 4.2(b) shows the shaped spectrum corresponding to the eye diagram of Figure 4.2(a). Observe that, due to the addition of the Manchester coding, a shallow notch in the spectrum has appeared at d.c. In the next section of the modulator, this notch will be enlarged to facilitate pilot insertion and pilot processing at the receiver.

After the pulse-shaping has been completed, the shaped I and Q streams are sent on to be highpass filtered. This is done to accentuate the notch at d.c., which was initialized by the Manchester coding. Two FIR highpass filters were considered for implementation: one with 91 taps and the other with 45. Close inspection of the resulting eye diagrams and spectra indicated that the 45 tap filter was sufficient for the desired purposes and was implementable within the DSP chip. The actual and scaled values of the 45 tap weights used in the TMS320 implementation are listed in table 4.2. The filter design specifications used in the computer-aided design of this filter are as follows:

Stopband:	0 to 100 Hz
Passband:	>=250 Hz
3 dB Point:	150 Hz
0.5 dB Ripple:	>=250 Hz
20 dB Attenuation at d.c.	

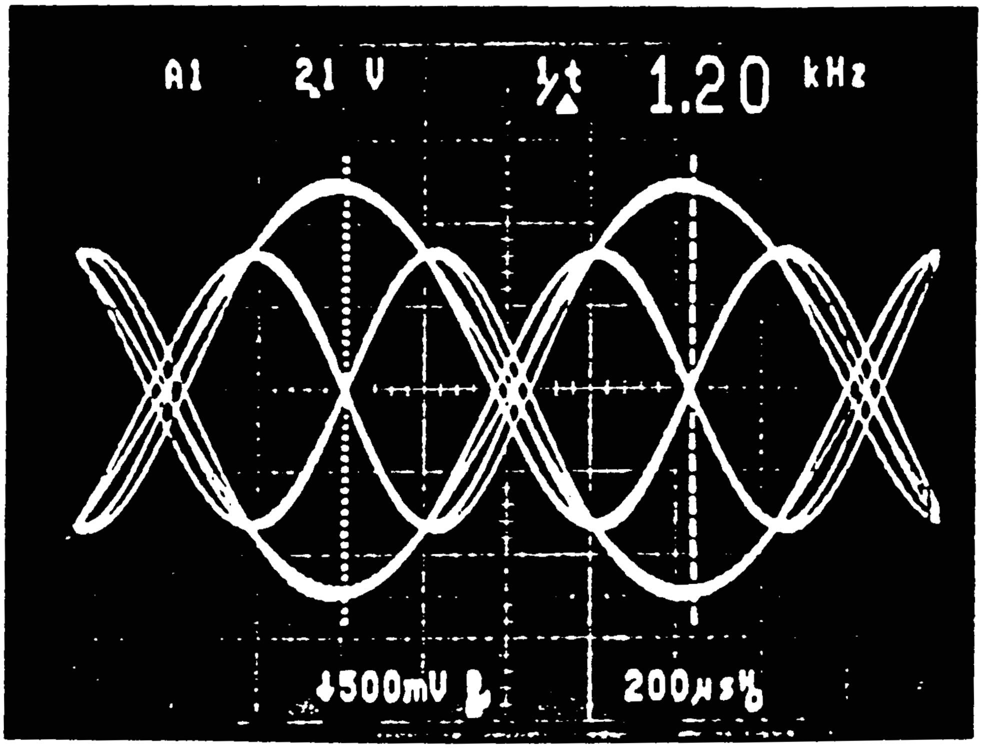
Linear Phase Response

The impulse reponse of the TMS320 version of this high pass filter is shown in Figure 4.3.

The pre-modulation processor output can be represented by the following constant coefficient difference equation:

$$x(k) = \sum_{\ell=0}^{44} s_k z^{-\ell} H(\ell) \tag{4.2}$$

ORIGINAL PAGE IS
OF POOR QUALITY



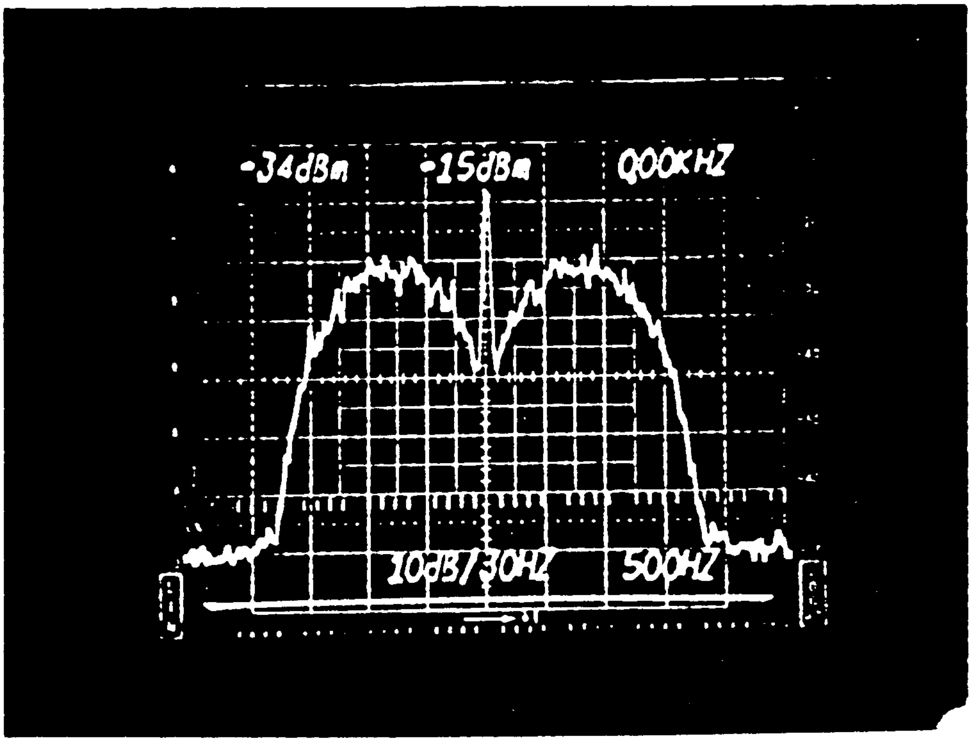
EYE DIAGRAM - 1.2 kbps

MANCHESTER CODING, 4 SAMPLES PER BIT SHAPING,

8-BIT DAC, BETA = 0.5

FIGURE 4.2(a)

ORIGINAL PAGE IS
OF POOR QUALITY



PULSE SHAPED SPECTRUM - 1.2 kbps

MANCHESTER CODING, 4 SAMPLES PER BIT SHAPING,

8-BIT DAC, BETA = 0.5

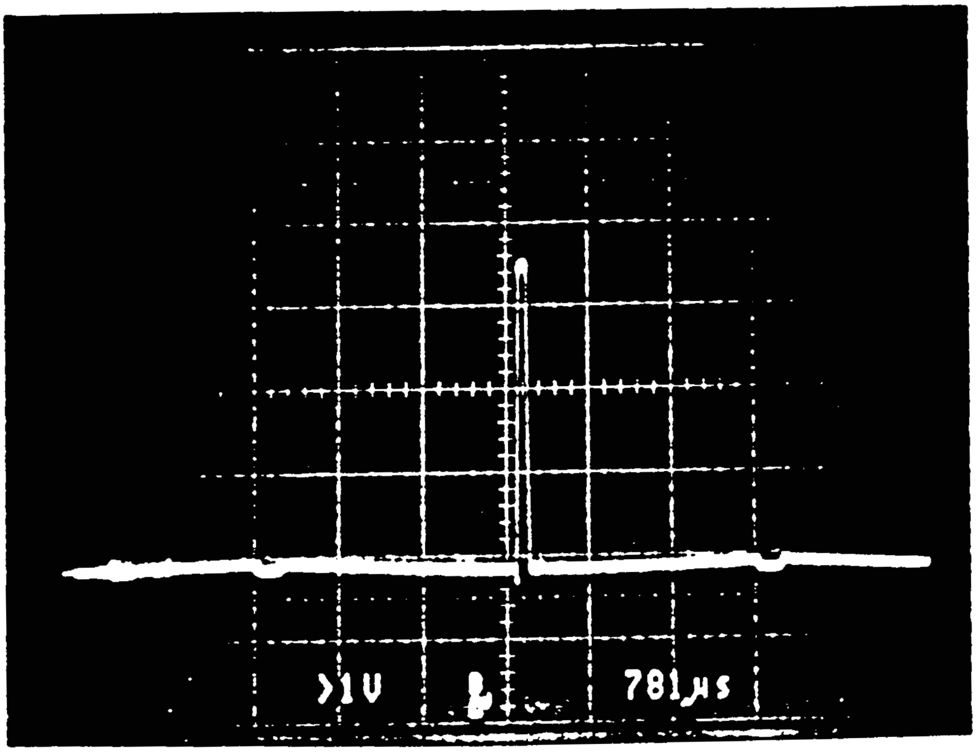
FIGURE 4.2(b)

Table 4.2

Tap Weights - 45 Order Filter

	Actual Value	Scaled Value
H(0) = H(44)	-.0281	-115
H(1) = H(43)	-.0083	-34
H(2) = H(42)	-.0094	-39
H(3) = H(41)	-.0106	-43
H(4) = H(40)	-.0118	-48
H(5) = H(39)	-.0130	-53
H(6) = H(38)	-.0142	-58
H(7) = H(37)	-.0155	-63
H(8) = H(36)	-.0167	-68
H(9) = H(35)	-.0179	-73
H(10) = H(34)	-.0191	-78
H(11) = H(33)	-.0202	-83
H(12) = H(32)	-.0213	-87
H(13) = H(31)	-.0223	-92
H(14) = H(30)	-.0232	-95
H(15) = H(29)	-.0241	-99
H(16) = H(28)	-.0249	-102
H(17) = H(27)	-.0255	-104
H(18) = H(26)	-.0261	-107
H(19) = H(25)	-.0265	-109
H(20) = H(24)	-.0268	-110
H(21) = H(23)	-.0270	-111
H(22)	.9829	3985

ORIGINAL PAGE IS
OF POOR QUALITY



IMPULSE RESPONSE

45 TAP HIGH PASS FILTER

FIGURE 4.3

where x(k) is the system output, $s_k = s(t_i + j)$ are the shaped samples from equation 4.1, and the H(scripl) terms are the high pass filter coefficients listed in Table 4.2. Highpass filtered inphase and quadrature samples are output simultaneously at a rate of 9.6 ksps. To generate the data staggering of OQPSK, the odd stream output is delayed by two output sample periods, or one-half of a Manchester bit period. Figure 4.4(a) shows the highpass filtered Manchester eye diagram. Note that intersymbol interference has been introduced at the sampling instants by the action of the highpass filter, as predicted in the software simulation. The frequency spectrum of the processed data for a random data input source is shown in Figure 4.4(b), and shows that the notch at d.c. has in fact been accentuated. However, this technique removes low frequency data energy as well, which results in the introduction of ISI into the transmit eye diagram.

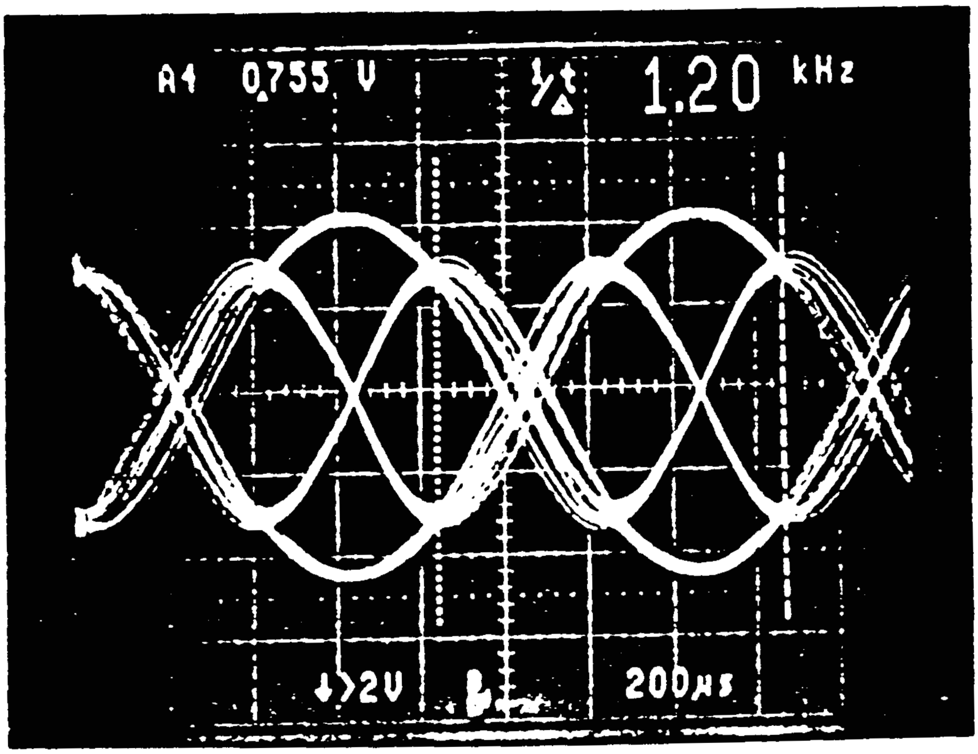
4.1.2 Demodulator

This section describes the final state of the hardware implementation of the MTCT baseband demodulator. As with the modulator, the main processing element in the demodulator is the TMS32010 DSP chip. The stand-alone digital demodulator circuitry has been designed at the schematic level and requires two TMS320 processor chips to implement in its entirety. The RF receiver circuitry, see [7], has been designed, built and tested. The demodulator code has been written for both TMS320 processors, and has been debugged to the extent possible with the available software simulation packages. The TMS320 demodulator software has been included in Appendix III.

A block diagram of the digital MTCT demodulator appears in Figure 4.5. It is necessary to employ two TMS320 processors due to the complexity of the demodulation scheme; the partioning of the signal processing requirements between the two processors is as indicated in Figure 4.5.

The received signal is translated to a suitable IF frequency by the RF circuitry and then converted to a digital signal prior to demodulation. The RF circuitry includes a step attenuator and bandpass filter which are used to accurately set carrier-to-noise ratios. The bandpass filter is also used to reject unwanted mixer products.

ORIGINAL PAGE IS
OF POOR QUALITY

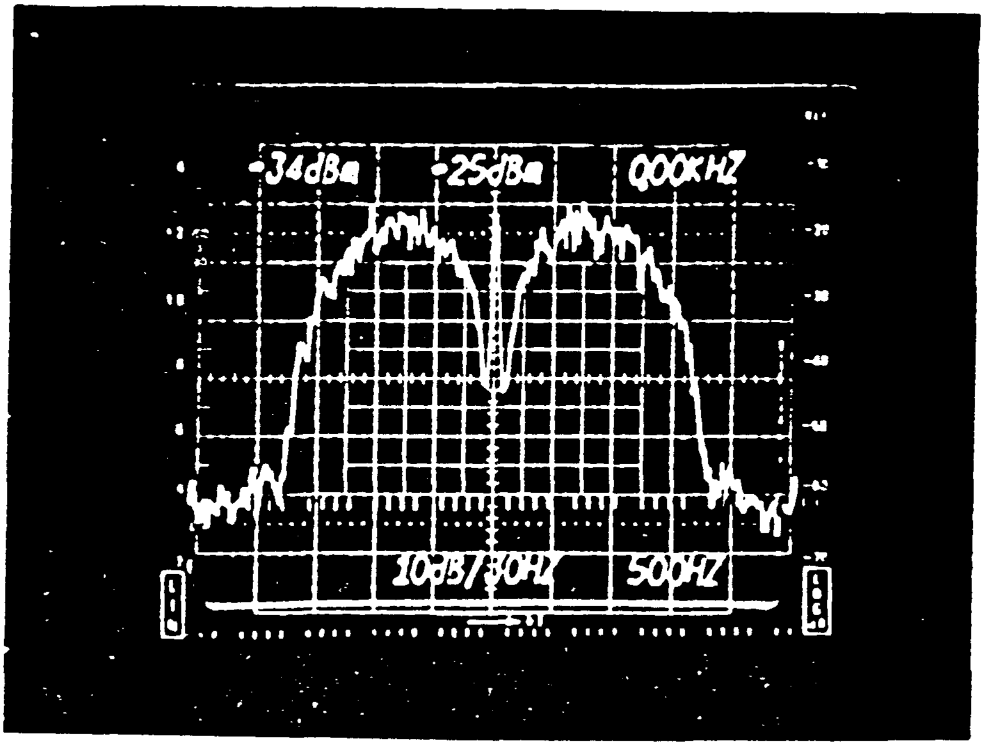


FILTERED EYE DIAGRAM - 1.2 kbps

MANCHESTER CODING, 4 SAMPLES PER BIT SHAPING,
8-BIT DAC, BETA = 0.5,
FOLLOWED BY A 45 TAP HIGH PASS FILTER

FIGURE 4.4(a)

ORIGINAL PAGE IS
OF POOR QUALITY



FILTERED PULSE SHAPED SPECTRUM - 1.2 kbps

MANCHESTER CODING, 4 SAMPLES PER BIT SHAPING,
8-BIT DAC, BETA = 0.5,
FOLLOWED BY A 45 TAP HIGH PASS FILTER

FIGURE 4.4(b)

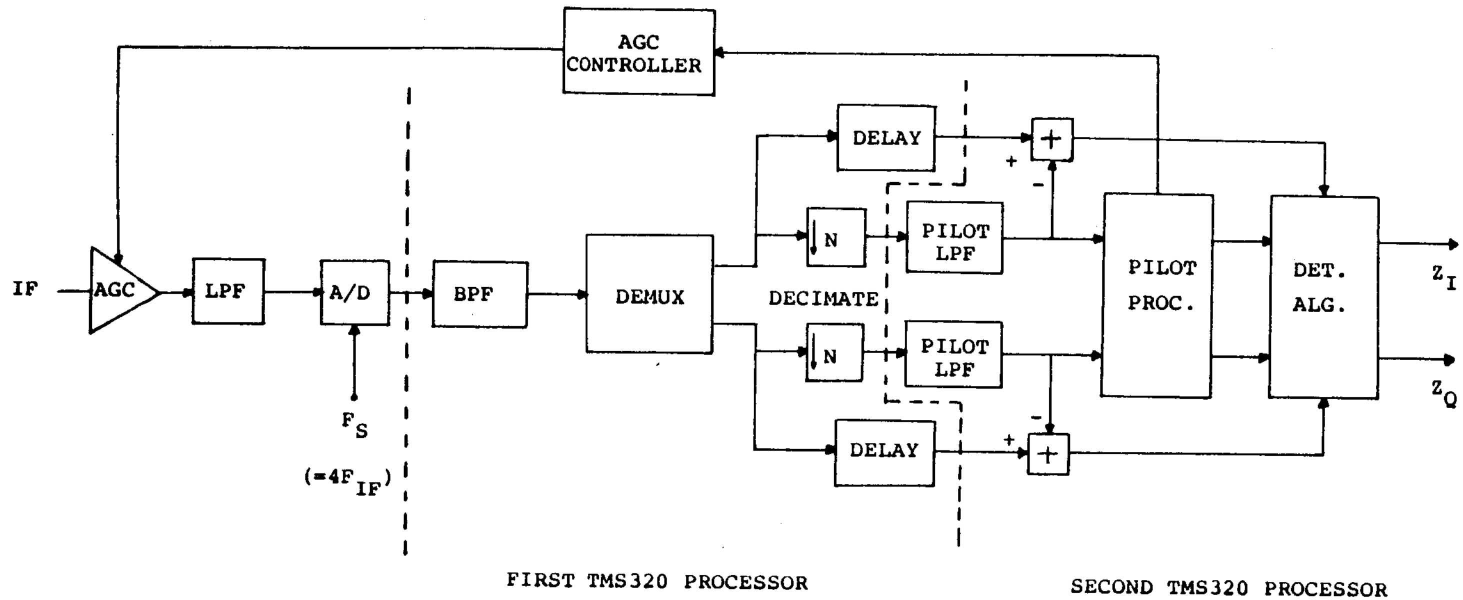


FIGURE 4.5 MTCT DEMODULATOR

The received signal is fed to the RF port of a double balanced mixer where it is mixed with the LO signal. A low-side mix arrangement is used with a 149.999988 MHz LO, producing an intermediate frequency of 12 kHz. The resulting IF signal is filtered and amplified by a fixed gain stage. Automatic Gain Control has not been considered in this implementation, due to its added complexity, however, if it was to be included, it would be introduced after the IF filtering as part of the IF gain stage. The output of the gain stage is then fed to the input of a 14-bit A/D converter, operating at 48 ksamples/second, i.e. four times the IF frequency, to generate the digital input to the baseband MTCT demodulator. This sample rate will eventually produced 10 samples per Manchester code bit or 20 per raw data bit, which, as shown by computer simulation, is more than sufficient for the demodulation and detection processes.

Due to this particular choice of sampling frequency, four times the intermediate frequency, it is apparent that quadrature sample pairs have been produced. This can be directly compared to the conventional generation of I and Q signals which employs quadrature analog mixing. As a direct result of the 4*IF sampling frequency, the sampled IF signal eminating from the A/D is such that every other pair of samples are phase inverted. Translation to baseband is accomplished by simply changing the sign of these inverted quadrature pairs. This is equivalent to mixing the IF signal with a square wave of the same frequency.

The amount of processing necessary for implementation of the digital demodulator cannot be performed at a rate of 48 ksamples per second by a single TMS320. To ease the implementation requirements, multirate processing has been used. This requires that the input signal be bandpass filtered to meet the Nyquist criterion for the maximum decimation signal processing path to avoid adjacent channel and noise aliasing (foldover). The bandpass filter meeting these requirements has the following specifications:

Passband Center:	12 kHz.
-3 dB points:	10.2 and 13.8 kHz.
-45 dB points:	6.24 and 17.74 kHz.

Linear Phase Response

The filter coefficients are listed in Table 4.3 along with the scaled values which are used in the TMS320 implementation. The bandpass filter code was tested using the TI TMS320 EVM (Evaluation Module) board; however, due to the limitation of the on-board A/D, the filter was implemented with a sample rate of 39 kHz. The amplitude and phase responses of the 39 kHz version of this bandpass filter are shown in Figures 4.6 and 4.7. These correspond to the desired 48 kHz sample rate filter when the above specifications are scaled by the factor (39/48).

The next step undertaken following the bandpass filter is to split the input data stream into inphase and quadrature channels. Recall that the I and Q streams are generated by sampling the IF signal at four times the IF frequency and processing the samples as discussed above. However, in the implementation considered, only every other sample pair is used thus avoiding the need for the inversion of alternate sample pairs without any loss of information. The remaining pairs are demultiplexed, with the first sample of each pair being placed in the I channel and the second in the Q channel. As a result of this demodulation and subsequent splitting of the received data stream, the rate of each of these channels is 12 ksamples per second, or 5 samples per Manchester code bit.

The remainder of the processing in the first TMS32010 is the same for both the inphase and quadrature channels and, therefore, only one channel will be described. The I (or Q) signal is first reproduced to form two duplicate streams. One of these duplicate streams is decimated by five, which reduces the sampling rate to 2.4 kHz, and is immediately sent on to the second TMS320 processor, where it will be low pass filtered as part of the pilot processing. It is important to note that this decimation does not cause aliasing in the

Table 4.3

Bandpass Filter Coefficients

	Actual Value	Scaled Value
H(1) = H(31)	-.1193 E-5	0
H(2) = H(30)	-.5156 E-3	-34
H(3) = H(29)	-.3013 E-6	0
H(4) = H(28)	-.01183	-775
H(5) = H(27)	-.2559 E-6	0
H(6) = H(26)	.0304	1994
H(7) = H(25)	-.7395 E-6	0
H(8) = H(24)	-.0302	-1980
H(9) = H(23)	.8667 E-6	0
H(10) = H(22)	-.0237	-1551
H(11) = H(21)	.1482 E-5	0
H(12) = H(20)	.1346	8822
H(13) = H(19)	-.8743 E-6	0
H(14) = H(18)	-.2524	-16544
H(15) = H(17)	.1010 E-5	0
H(16) =	.3035	19888

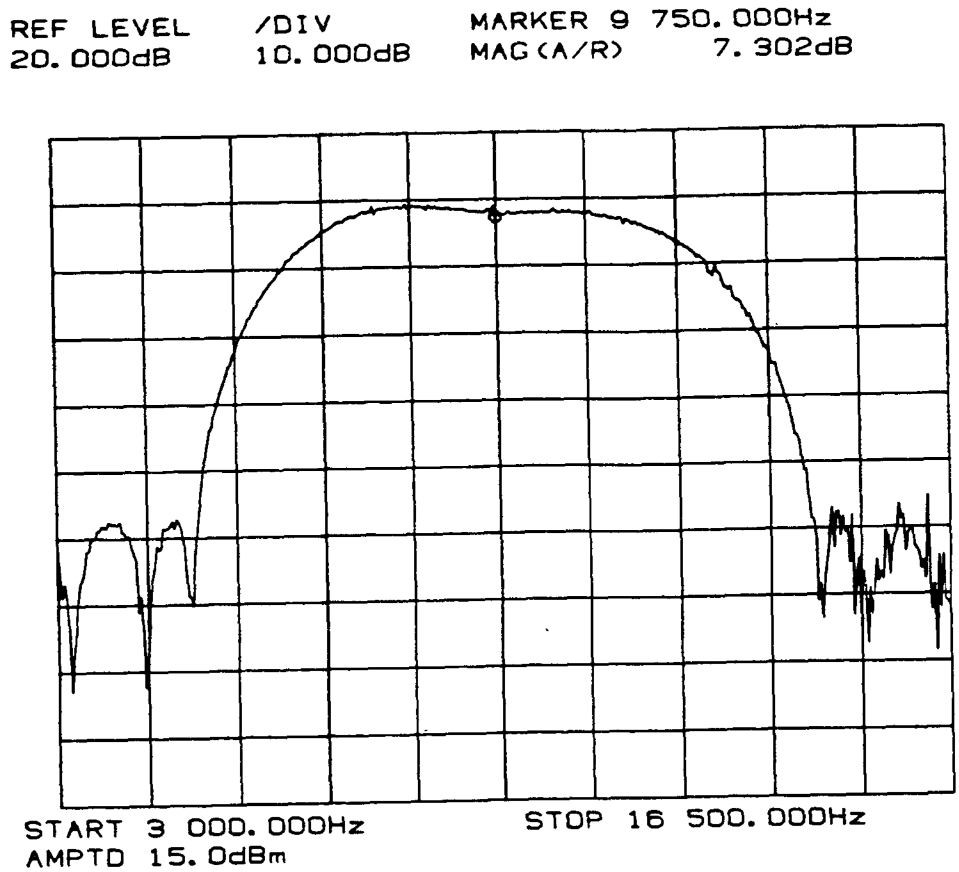


FIGURE 4.6 MAGNITUDE RESPONSE OF BANDPASS FILTER, 16 BIT COEFFICIENTS, SAMPLING FREQUENCY = 39 kHz

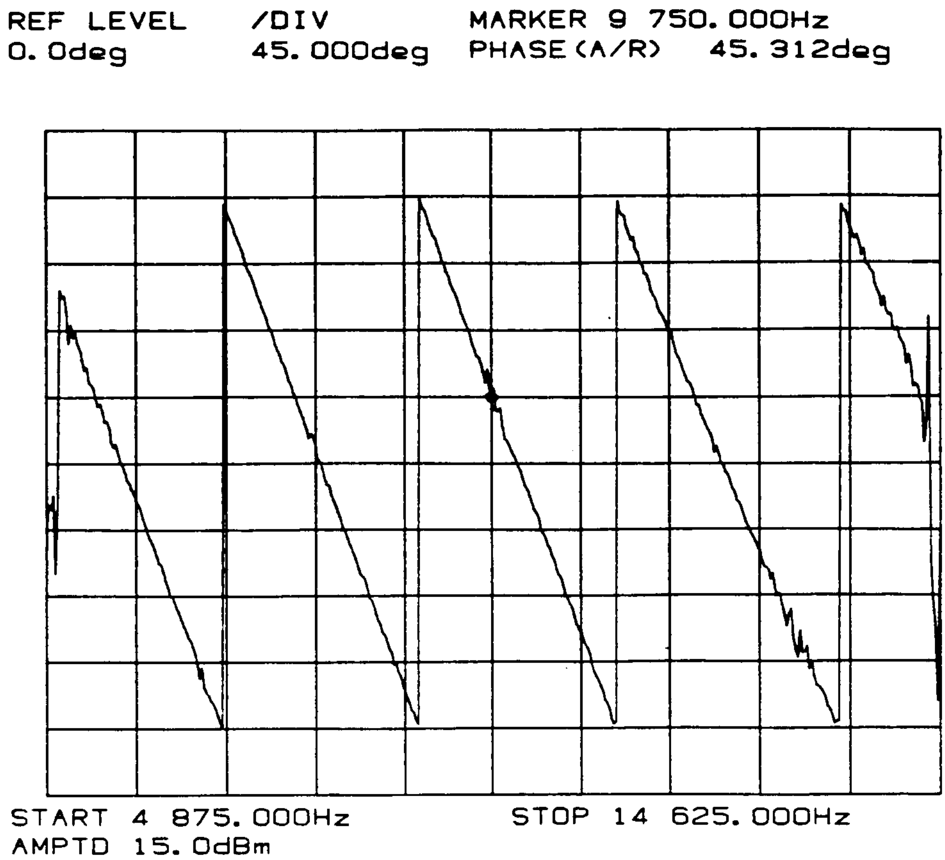


FIGURE 4.7 PHASE RESPONSE OF BANDPASS FILTER IN FIGURE 4.6

pilot filter passband but does improve the filter cut-off to sampling ratio, making it possible to use a lower order filter.

The other sample stream is placed into an external RAM, which functions as a delay buffer to compensate for the pilot processing input/ouput delay, thereby maintaining synchronization between the two streams. The external RAM stores the samples until it is time to send them on to the second TMS320 processor, where the pilot tone component will be removed from these appropriately delayed samples. The remaining functions of the demodulator are performed in the second processor which, conceptually, consists of two parallel paths, one dedicated to the pilot processing and the other to the delayed data. These paths converge at the detection algorithm, where fade compensation and synchronous data detection are performed. As stated above, the I and Q data streams undergo identical processing, as do the inphase and quadrature baseband pilot components.

The first operation performed in the processing of the pilot is to lowpass filter the undelayed, decimated I and Q sample trains in order to recover the pilot tone components. Assuming an expected worst case fading frequency of 80 Hz at the receiver, the lowpass filter design specifications are as follows:

-3 dB at 80 Hz.
-39 dB at 160 Hz.

Linear Phase Response

The pilot lowpass filter coefficients are listed in table 4.4. Included are the 13 bit scaled integer values used in the TMS320 software. The pilot digital filter was designed to operate at 2.4 kHz to achieve its desired frequency response; however, it was tested on the TMS320 EVM board at a 39 kHz sampling rate for better frequency resolution. The resulting magnitude and phase responses, of Figures 4.8 and 4.9, are therefore scaled by the factor (39/2.4). Responses are not shown to zero frequency due to leakage from the local oscillator of the network analyzer.

Table 4.4

Pilot Lowpass Filter Coefficients

	Actual Value	Scaled Value
H(0) = H(64)	-.00623	-204
H(1) = H(63)	-.00079	-26
H(2) = H(62)	-.00031	-10
H(3) = H(61)	.00052	17
H(4) = H(60)	.00164	54
H(5) = H(59)	.00294	96
H(6) = H(58)	.00426	139
H(7) = H(57)	.00538	176
H(8) = H(56)	.00609	200
H(9) = H(55)	.00619	203
H(10) = H(54)	.00550	180
H(11) = H(53)	.00393	129
H(12) = H(52)	.00148	49
H(13) = H(51)	-.00173	-57
H(14) = H(50)	-.00550	-179
H(15) = H(49)	-.00936	-307
H(16) = H(48)	-.01297	-425
H(17) = H(47)	-.01578	-517
H(18) = H(46)	-.01728	-566
H(19) = H(45)	-.01696	-556
H(20) = H(44)	-.01443	-473
H(21) = H(43)	-.00945	-310
H(22) = H(42)	-.00193	-63

Table 4.4

Pilot Lowpass Filter Coefficients

	Actual Value	Scaled Value
H(23) = H(41)	.00799	262
H(24) = H(40)	.01994	654
H(25) = H(39)	.03338	1094
H(26) = H(38)	.04758	1559
H(27) = H(37)	.06169	2021
H(28) = H(36)	.07483	2452
H(29) = H(35)	.08613	2822
H(30) = H(34)	.09480	3107
H(31) = H(33)	.10028	3286
H(32)	.10215	3347

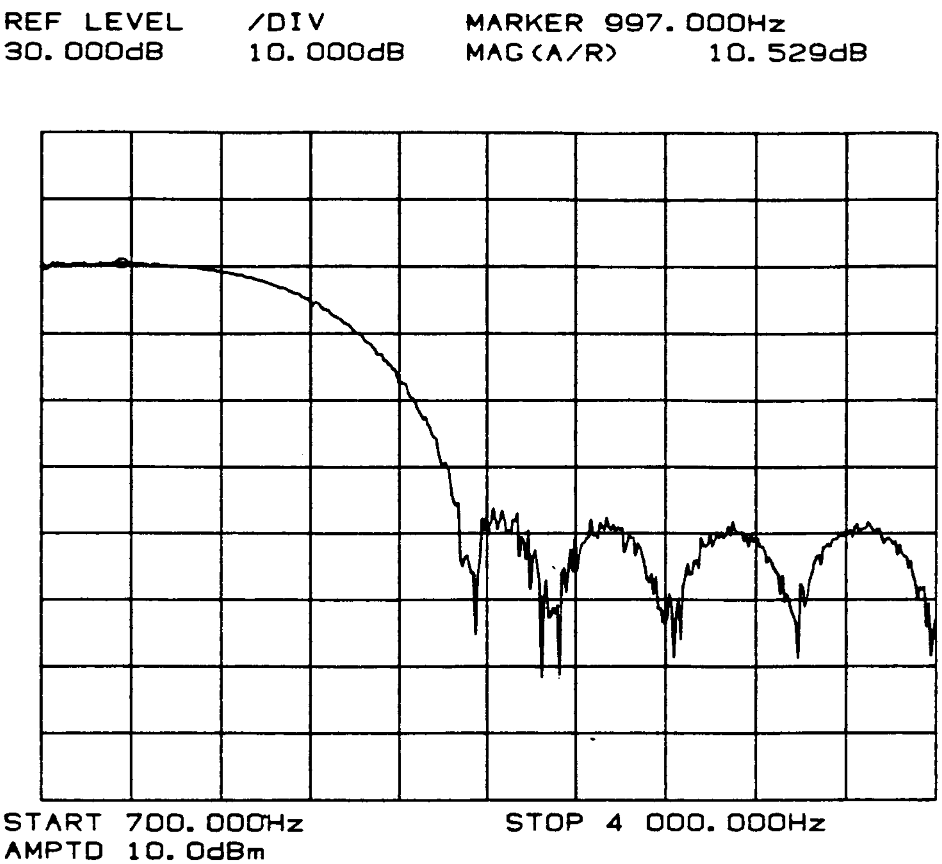


FIGURE 4.8 MAGNITUDE RESPONSE OF THE PILOT LOWPASS FILTER, 13 BIT COEFFICIENTS, SAMPLING FREQUENCY = 39 kHz

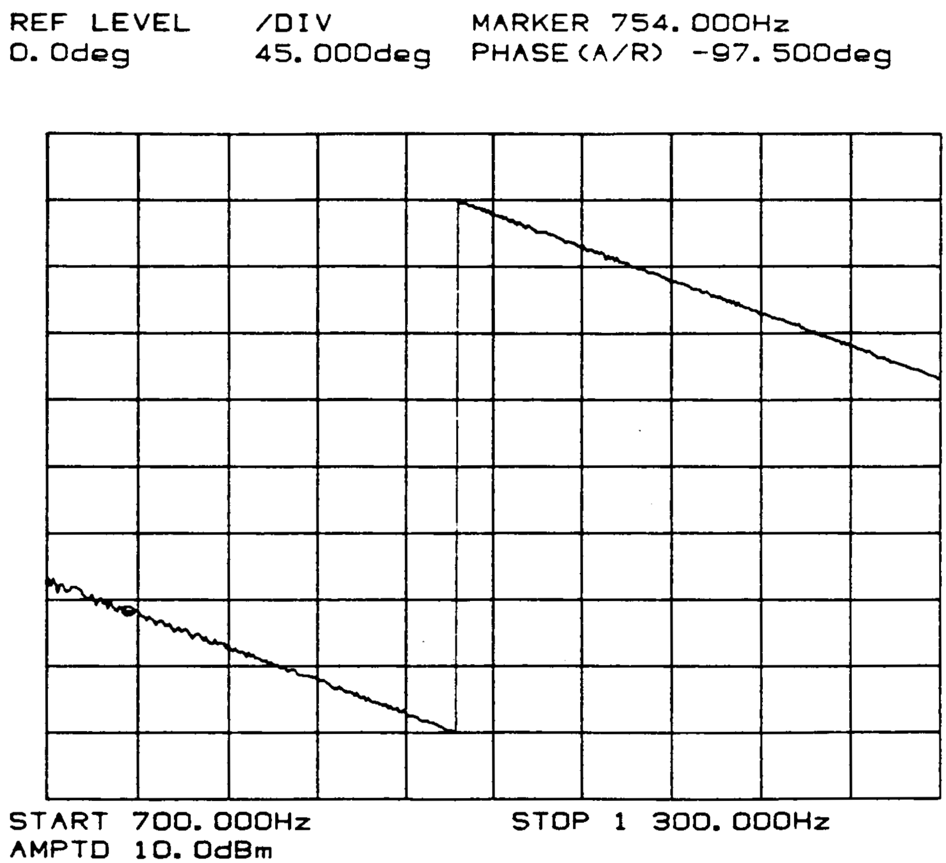


FIGURE 4.9 PHASE RESPONSE OF THE PILOT LOWPASS FILTER IN FIGURE 4.8

The filtered I and Q streams are utilized in two ways: (1) they are passed on to the pilot processor and (2) they are subtracted from their respective delayed I/Q sample train. The latter operation removes the pilot from the unfiltered 12 kHz sample stream and the result is sent to the detection algorithm.

The subtraction of the lowpass filtered signal from the delayed sample stream may at first appear to be straightforward, however it is complicated by the multirate processing. Specifically, the unfiltered stream has a sample rate of 12 kHz, while the filtered stream, having been decimated by five, has a sample rate of only 2.4 kHz. In order to perform the subtraction, a linear interpolation of the sample values of the 2.4 kHz stream is used to raise the sample rate. Linear interpolation was selected over the less complex zero-order hold method because the software simulation of this demodulator indicated that the additional accuracy afforded by the interpolation method is necessary.

The I and Q recovered pilot components, at 2.4 ksamples/sec, are passed onto the pilot processor block which extracts the fading phase information. This, in turn, is passed onto the detection algorithm where it is used to mitigate the fading effects on the data sidebands. This process is based on the assumption that the pilot has been exposed to the same channel perturbations as the data, e.g. Rician or Rayleigh multipath fading. The I and Q outputs of the the pilot processor are used in the detection algorithm as coherent phase references. The pilot processing function is detailed in equation 4.3.

$$\tan^{-1}(\phi) = Q_p/I_p = \cos(\phi)/\sin(\phi) \tag{4.3}$$

In the actual implementation, the inverse tangent is not evaluated, rather, the result of the Q_p/I_p division is used to determine the corresponding sine and cosine values via a look-up table. The periodicity and symmetry of the trigonometric functions are exploited to minimize the size of this look-up table, only $\cos(\phi)$ and $\sin(\phi)$ values for ϕ in the range $[0,\pi/4]$ need be stored in order to implement this function. Therefore, this processing block will first divide a quadrature sample by its corresponding inphase sample, then simply use this look-up table to locate the appropriate cosine and sine values. The $[0,\pi/4]$ angle range is subdivided into 128 levels. Software simulations have shown that this is of sufficient accuracy. These cosine and sine terms are sent on to the next phase in the demodulator, the detection algorithm.

The last section of the demodulator is the detection algorithm. This function block uses the sine and cosine of the pilot angle, which at this point should consist of the pilot tone with amplitude variations removed, as coherent phase references for the simultaneous operations of data recovery and the removal of channel phase perturbations. The algorithm is as follows:

$$Z_I = 2I_D\cos(\phi) + 2Q_D\sin(\phi) \tag{4.4}$$

and

$$Z_Q = 2Q_D\cos(\phi) - 2I_D\sin(\phi) \tag{4.5}$$

where I_D and Q_D are the inphase and quadrature samples from the 12 kHz streams. The demodulated signals, Z_I and Z_Q, are then fed off chip to analog integrate-and-dump data detectors to produce an estimate of the transmitted data.

As in the case of the removal of the pilot signal from the received signal, the detection process is complicated by the multirate processing. Here, the I and Q streams are sampled at 12 kHz while the cosine and sine signals are sampled at only 2.4 kHz. As before, linear interpolation is employed to match the rates of the two streams. The outputs of the detector consist of one inphase and one quadrature data channel, which are filtered to remove out-of-band noise before they are sent on to the detector/decoder board. The specifications for this final data filter are

-3 dB at 2.0 kHz.
-30 dB at 2.4 kHz.

Linear Phase Response

The coefficients and their scaled integer values for the data lowpass filter are listed in table 4.5. The designed sampling frequency is 12 kHz. The tested sampling frequency is 19.52 kHz, yielding a scaling factor of (19.52/12) for the magnitude and phase frequency responses of Figures 4.10 and 4.11.

4.2 Subcarrier TCT Modulator

In section 2, the subcarrier method of TCT was presented, and proved to be similar, in many respects, to the Manchester encoded TCT. The basic difference between the two approaches is the way in which they create the spectral null at d.c. The STCT version creates this null by simply modulating the data onto a very low frequency subcarrier. The similarities between the two approaches carry through to their hardware implementation. A hardware version of the STCT modulator was also completed. It utilizes the same stand-alone board and RF circuitry employed in the MTCT modulator. In order to change modulation methods, it is only necessary to replace the two EPROM's used for program memory with those containing the subcarrier code.

The TMS320 implementation of the STCT modulator, outlined previously in Figure 2.3, is quite similar to the MTCT version described previously. The data is input at the same rate, 2.4 kbps, and is immediately split into separate inphase and quadrature channels. The Manchester coding is replaced here by a simpler bipolar coding scheme, and the code bits are then sent on at a rate of 1.2 kbps to the pulse shaping section.

The pulse shape used for the STCT modulator is the same as that employed in the MTCT modulator, a raised-cosine waveform with a β of 0.5, truncated to eight code bits. The subcarrier version, however, represents this waveform in the digital domain by eight samples per bit, instead of the four which were

Table 4.5

Data Lowpass Filter Coefficients

	Actual Value	Scaled Value
H(0) = H(40)	-.00453	-37
H(1) = H(39)	-.00306	-25
H(2) = H(38)	.01315	108
H(3) = H(37)	-.00009	-1
H(4) = H(36)	-.00535	-44
H(5) = H(35)	-.01099	-90
H(6) = H(34)	-.00104	-8
H(7) = H(33)	.01268	104
H(8) = H(32)	.01460	120
H(9) = H(31)	-.00315	-26
H(10) = H(30)	-.02296	-188
H(11) = H(29)	-.01866	-153
H(12) = H(28)	.01308	107
H(13) = H(27)	.03960	324
H(14) = H(26)	.02212	181
H(15) = H(25)	-.03587	-294
H(16) = H(24)	-.07452	-610
H(17) = H(23)	-.02444	-200
H(18) = H(22)	.12216	1001
H(19) = H(21)	.28662	2348
H(20)	.35858	2937

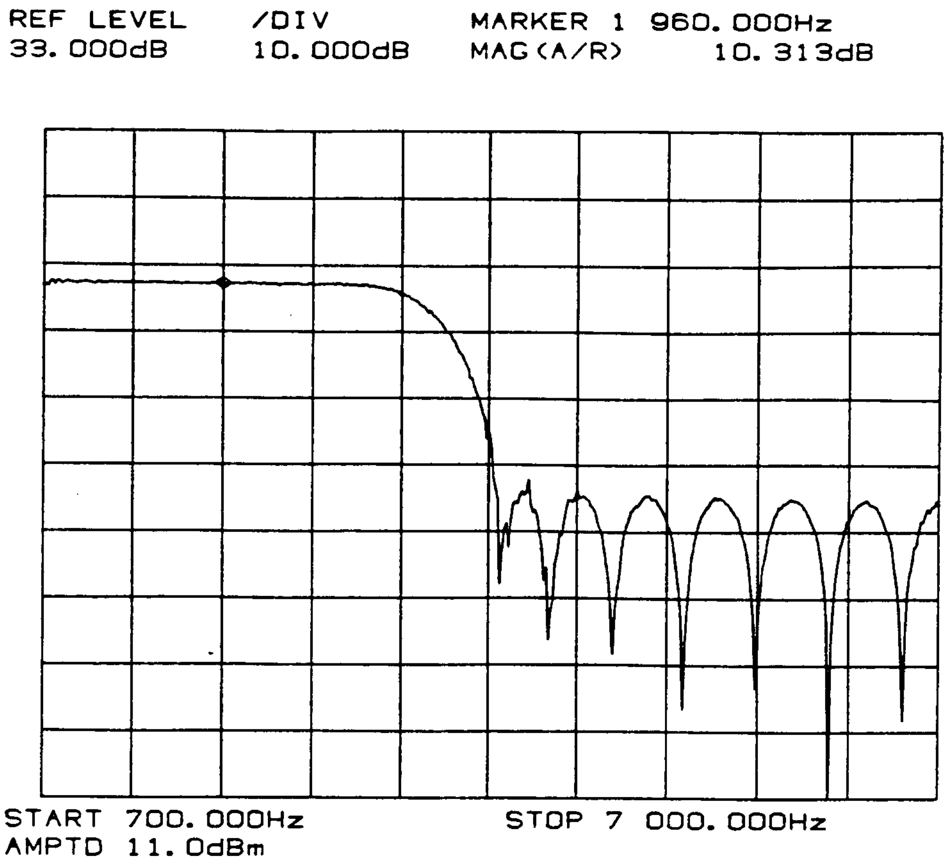


FIGURE 4.10 MAGNITUDE RESPONSE OF THE DATA LOWPASS FILTER

13 BIT COEFFICIENTS, SAMPLING FREQUENCY = 19.52 kHz

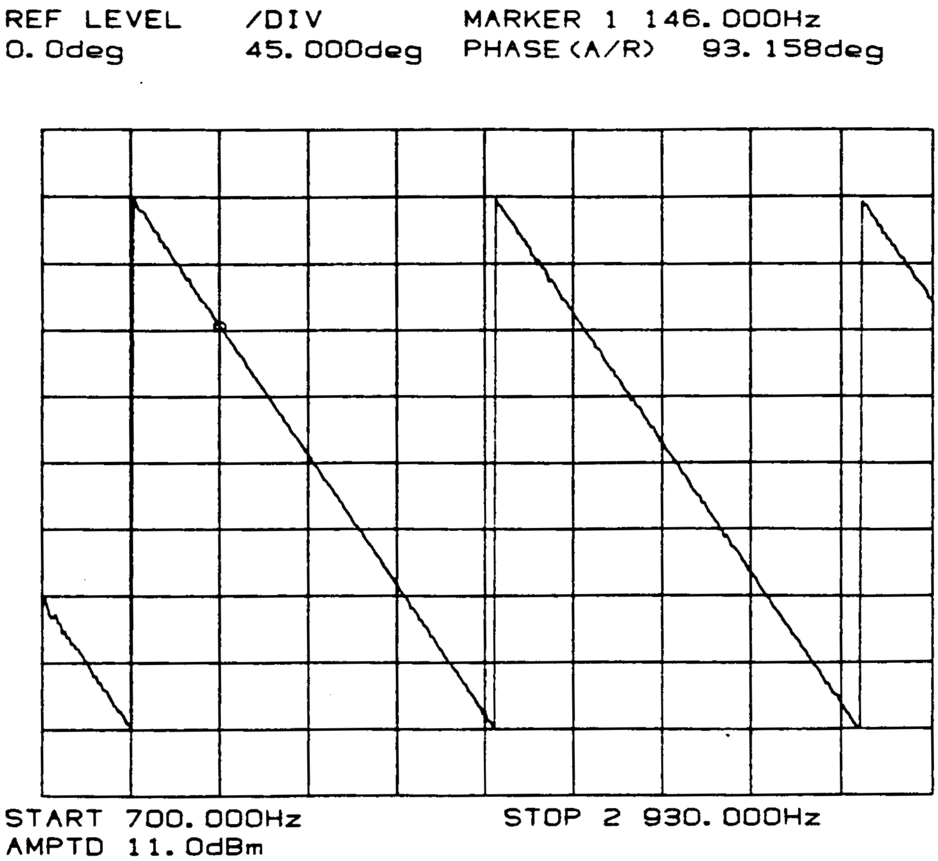


FIGURE 4.11 PHASE RESPONSE OF THE DATA LOWPASS FILTER IN FIGURE 4.10

used earlier. This doubling of the number of samples per bit insures that the output of the pulse shaping section has a rate of 9.6 ksamples per second, the same rate the output waveform has in the MTCT implementation.

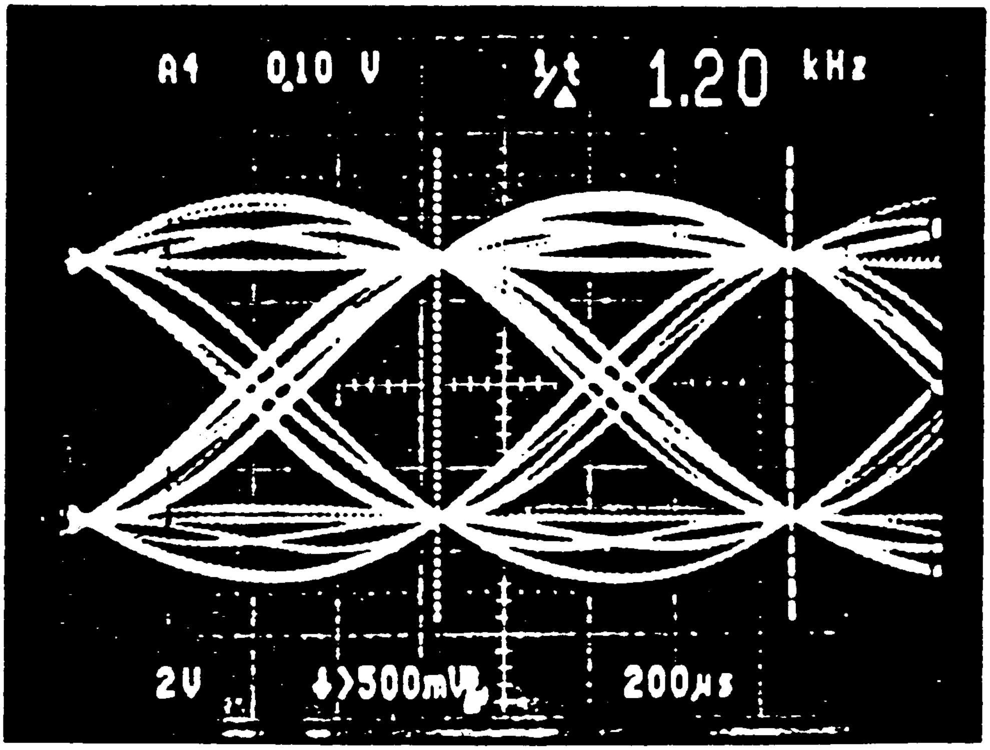
The eye diagram produced by the STCT pulse shaping section is shown in Figure 4.12(a). Note, once more, that at the sampling instants there is negligible intersymbol interference. The spectrum of this waveform is shown in the following illustration, Figure 4.12(b), and is the required 40 dBc at 1.8 kHz.

The final block of the STCT processor is the modulation section. Here, the shaped even stream is multiplied by a cosine term, the odd stream by a sine term, they are summed and then passed to the RF circuitry. The subcarrier frequency of these sine and cosine terms was chosen to be 960 Hz because it is an integer submultiple of the data rate 9.6 kHz. This means that only ten sample values of each sinusoid need to be stored in order to implement the modulator. The result of this modulation is then sent on to the D/A on the RF board.

The spectrum produced by the STCT modulator is shown in Figure 4.13. The modulation has produced a deep null at zero frequency, where the pilot would be placed. This null is deeper than that created by the MTCT modulator, as expected. The null width could be increased and the bandwidth reduced to the specified 3.6 kHz by simply changing the pulse shape to have a β value of 0.4.

5. CONCLUSIONS

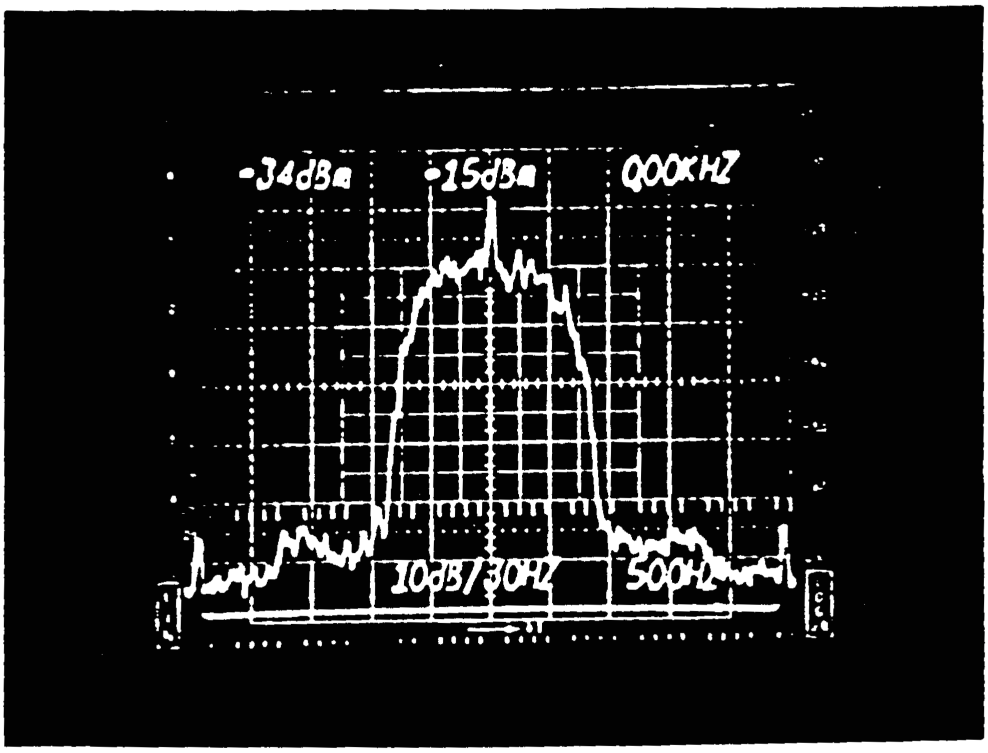
The two major goals of this program were the design of an improved, all digital, Manchester encoded based TCT modulator as well as the investigation of a baseband I/Q demodulator and detector. It is believed that both of these goals were achieved and, in addition, a viable alternative to the Manchester system, the subcarrier technique, was derived.



EYE DIAGRAM - 1.2 Kbps

NRZ CODING, 8 SAMPLES PER BIT SHAPING,
8 BIT DAC, BETA = .5

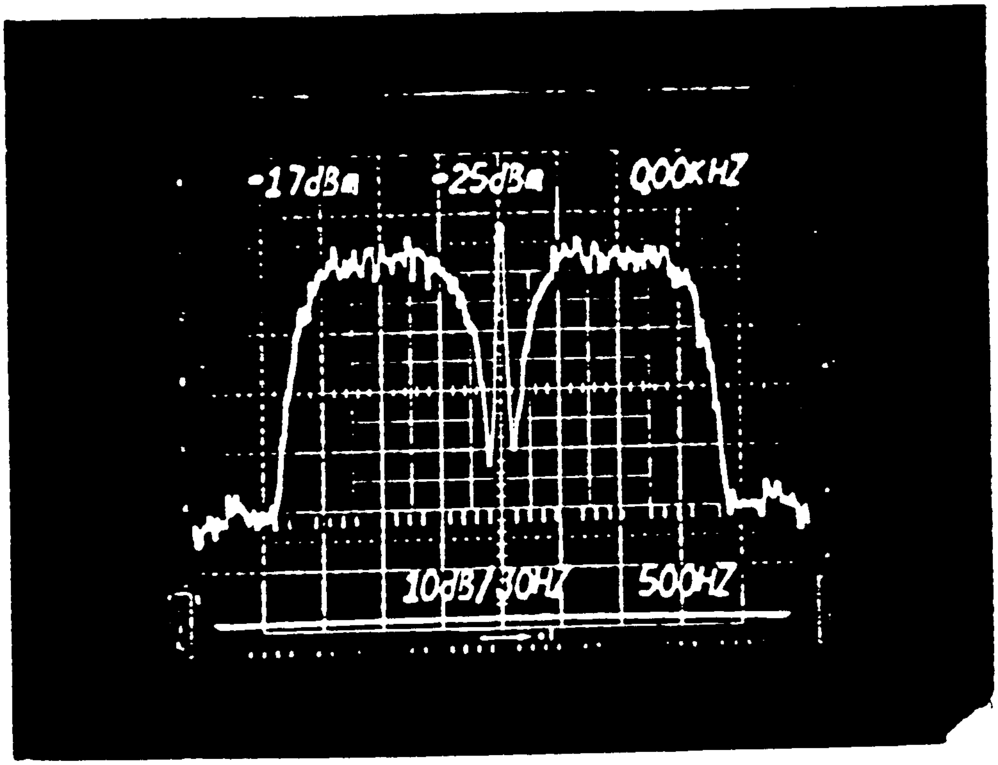
FIGURE 4.12(a)



PULSE SHAPED SPECTRUM - 1.2 Kbps

NRZ CODING, 8 SAMPLES PER BIT SHAPING,
8 BIT DAC, BETA = .5

FIGURE 4.12(b)



MODULATED PULSE SHAPED SPECTRUM - 1.2 Kbps

NRZ CODING, 8 SAMPLES PER BIT SHAPING,
8 BIT DAC, BETA = .5

FIGURE 4.13

Computer simulation and hardware implementation were employed to investigate the Manchester TCT modulator proposed by Davarian [2] which included a highpass filter to improve the spectral null created by the Manchester source encoding. The results obtained showed that the spectral null at zero frequency could be enlarged by the filtering, however, the removal of the low frequency data energy introduced intersymbol interference of approximately 18% into the transmit data eye. This was considered to be a significant disadvantage given the amount of additional filtering that was required in both the inphase and quadrature paths. It is apparent that this technique is less than optimum especially since no advantage was being derived from the raised-cosine shaping in the critical area of the spectral null.

To gain the advantage of the pulse-shaping and simultaneously remove the need for the highpass filters, it was clear that a subcarrier modulation technique should be explored. This would permit the arbitary location of the upper and lower data sidebands at a point where they would be symmetrical around the transmit band center. This would also allow for equal sideband roll-off without incurring an ISI penalty, by virtue of the excess bandwidth fraction. In this way the shape of the data spectrum around zero frequency can be easily controlled. In addition, the premodulation processing is simplified, as was shown in section 2.2.1. It has also been demonstrated that, using the STCT processor, it is now possible to perform fully digital QPSK modulation with all the attendant advantages, such as improved carrier suppression, pilot insertion and adjustment free operation. The subcarrier TCT modulator was simulated and constructed, and demonstrated superior performance to that of the MTCT counterpart. The use of the subcarrier, however, slightly complicates the demodulator arrangement over that of the MTCT system but this is not considered to be a serious problem, as has been borne out by computer simulation.

Considerable effort was directed towards the design, computer simulation and implementation of a baseband TCT compatible demodulator. The salient features of the selected configuration are: a pilot phase-recovery-only scheme, used to reduce implementation complexity; inband pilot components in the I and Q data paths, removed by a simple subtraction process; multi-rate processing, also for reduced complexity; and a provision for a long term AGC function.

Computer simulation of both the Manchester and subcarrier demodulators, in conjunction with their respective modulators, revealed no conceptual problems, however, neither system was tested with either simulated noise or fading. The results of the software simulation did show that for a 16 bit processor architecture, the demodulator processing should not significantly degrade the overall system performance. This was confirmed by preliminary results of the real-time implementation. Direct comparisons of digital filter frequency responses between the simulation and the hardware indicated little difference in both magnitude and phase. As a result, the increase in complexity for floating point arithmetic processing is not considered an acceptable alternative. The same reasoning applies to any discrete hardware approach. The TMS320 provides sufficient processing power and the shortest critical path time in system development.

The additional processing required by the subcarrier demodulator, the remodulation and phase estimation processes, do not appear to impact the system performance and are readily implementable in the DSP chip. The simulations show that the subcarrier phase estimation loop acquires sychronization rapidly, as would be expected from a first order loop, and, consequently, would have little impact on the system throughput.

Given the RF channel allocation and eventual data rate of 4.8 kbps in 5 kHz, it would appear that neither Manchester nor the subcarrier techniques offer a viable solution. Even with expected peformance gain of the subcarrier technique, it is clear that the system is very wasteful in terms of bandwidth and as a result requires excessively large M-ary signalling schemes for data transmission.

A potential candidate TCT scheme which approaches the system bandwidth goals is the dual-pilot method proposed by General Electric [8], and subsequently analyzed by Simon [9]. In this scheme, the bandwidth requirements can be reduced by a factor of two, however, there is a power performance penalty incurred due to the use of two pilots. This penalty may tend to ultimately balance out the performances of the single and dual pilot schemes.

The dual-pilot scheme is slightly more complicated that either single pilot scheme but is still amenable to digital implementation using similar techniques to those developed during this program. Consequently, this scheme would be a worthwhile subject for future work in an attempt to derive the optimum TCT transceiver for the satellite fading mobile communication link.

6. REFERENCES

[1] General Electric Corporate Research and Development, "A Study of Tone Calibrated Technique", Vol. 1, Technical Proposal Prepared for JPL RFP No. OH-6-6960.

[2] F. Davarian, "High Performance Communication in Mobile Channels", IEEE 34th Vehicular Technology Conference, Pittsburgh, Pa. Session C5, May 1984.

[3] J. McGeehan, A. Bateman, "Theoretical and Experimental Investigation of Feedforward Signal Regeneration", IEEE Trans. Vehicular Technology, Vol. VT-32, pp. 106-120, Feb. 1983.

[4] F. Davarian, M. Simon, J. Sumida, "DMSK: A Practical 2400 bps Satellite Receiver for Mobile Satellite Service", JPL Publication 85-51, MSAT-X Report No. 111, June 15,1985.

[5] General Electric Corporate Research and Development, "An Additional Study and Implementation of Tone Calibrated Technique of Modulation - First Interim Report", Prepared for JPL Contract No. 957190, August 1985.

[6] Jet Propulsion Lab, "Baseband Implementation of a Tone Calibrated Receiver", Exhibit I, JPL Contract No. 957190, June 1985.

[7] General Electric Corporate Research and Development, "An Additional Study and Implementation of Tone Calibrated Technique of Modulation - Second Interim Report", Prepared for JPL Contract No. 957190, August 1985.

[8] General Electric Corporate Research and Development, "Design of a MSAT-X Mobile Transceiver and Related Ground Segment Technology", Technical Proposal to JPL, GE No. 1C-6-0357-215, September 1984.

[9] M. K. Simon, "Dual Pilot Tone Calibration Technique (DPTCT)", Internal JPL Document.

APPENDIX I

TMS320 BANDPASS FILTER SOFTWARE

```
                 IDT      'BPF'
             ***********************************************************
             *                                                         *
             *      SOFTWARE FOR BOARD 1 OF THE REAL TIME              *
             *      IMPLEMENTATION OF THE MANCHESTER TCT               *
             *                 DEMODULATOR                             *
             *                                                         *
             *      Written by - Norman E. Lay                         *
             *      Last Updated : 8/30/85                             *
             *                                                         *
             *      General Electric Company                           *
             *      Corporate Research & Development                   *
             *      Schenectady, N.Y.                                  *
             *                                                         *
             *---------------------------------------------------------*
             *                                                         *
             *      The following TMS-320 assembler code               *
             *      implements a bandpass filter at a samp-            *
             *      ling frequency of 48kHz with a center              *
             *      frequency of 12kHz and a passband width            *
             *      of 3.6kHz. Additional overhead functions           *
             *      are also performed, including decimation           *
             *      by 4:1 (data) and by 20:1 (pilot) and de-          *
             *      lay equalization to compensate for the             *
             *      pilot lowpass filter in the demodulator.           *
             *      The output of this board will be 2 12kHz           *
             *      data streams, 2 2.4kHz pilot streams and           *
             *      a synchronization pulse to align the pilot         *
             *      and data streams.                                  *
             *                                                         *
             *---------------------------------------------------------*
             *
       0000  INPUT  EQU      >0      )
       0001  Z1     EQU      >1      ) Beginning of Ram
       0002  Z2     EQU      >2      ) for Delay Storage
       0003  Z3     EQU      >3      )        .
       0004  Z4     EQU      >4      )        .
       0005  Z5     EQU      >5      )        .
       0006  Z6     EQU      >6      )        .
       0007  Z7     EQU      >7      )
       0008  Z8     EQU      >8      )
       0009  Z9     EQU      >9      )
       000A  Z10    EQU      >A      )
       000B  Z11    EQU      >B      )
       000C  Z12    EQU      >C      )
       000D  Z13    EQU      >D      )
       000E  Z14    EQU      >E      )
       000F  Z15    EQU      >F      )
       0010  Z16    EQU      >10     )
       0011  Z17    EQU      >11     )
       0012  Z18    EQU      >12     )
       0013  Z19    EQU      >13     )
       0014  Z20    EQU      >14     )
       0015  Z21    EQU      >15     )
       0016  Z22    EQU      >16     )
       0017  Z23    EQU      >17     )
       0018  Z24    EQU      >18     )
```

```
     0019 Z25    EQU   >19     )      .
     001A Z26    EQU   >1A     )      .
     001B Z27    EQU   >1B     )      .
     001C Z28    EQU   >1C     )      .
     001D Z29    EQU   >1D     ) End of Ram for
     001E Z30    EQU   >1E     ) Delay Storage
          *
     001F TAP2   EQU   >1F     ) Beginning of Ram
     0020 TAP4   EQU   >20     ) Storage for Coeffs.
     0021 TAP6   EQU   >21     )      .
     0022 TAP8   EQU   >22     )      .
     0023 TAP10  EQU   >23     )      .
     0024 TAP12  EQU   >24     )      .
     0025 TAP14  EQU   >25     ) Half of the filters
     0026 TAP16  EQU   >26     ) coefficients are
     0027 TAP18  EQU   >27     ) coded as zeros
     0028 TAP20  EQU   >28     )      .
     0029 TAP22  EQU   >29     )      .
     002A TAP24  EQU   >2A     )      .
     002B TAP26  EQU   >2B     )      .
     002C TAP28  EQU   >2C     ) End of Ram Storage
     002D TAP30  EQU   >2D     ) for Coefficients
          *
     003E BPF1   EQU   >3E        ) Input Buffer to
     003F BPF2   EQU   >3F        ) Delay.
     0040 PILOT2 EQU   >40     )
     0041 PILOT1 EQU   >41     ) Output Buffers to 2nd
     0042 DBPF1  EQU   >42     ) Demodulator Board
     0043 DBPF2  EQU   >43     )
     0044 TWO    EQU   >44        ) Constants = 1,2
     0045 ONE    EQU   >45        )
     0046 OUTFLG EQU   >46     ) Decimation and Sync.
     0047 SNCFLG EQU   >47     ) Flags
     0048 DLYADX EQU   >48        ) External Ram Address Pointer
     0049 DLYBEG EQU   >49     )
     004A FIFOND EQU   >4A     ) Stored Address Constants
          *
     007E IACH   EQU   >7E     ) Accumulator Storage
     007F IACL   EQU   >7F     ) During an Interrupt
          *
     0340 DLYSIZ EQU   >340    ) MPYK Constant
     0200 BUFBEG EQU   >200    ) Definitions
     0400 COEFFS EQU   >400    )
          *
          *      Begin TMS-320 Code
          *
0000             AORG  >0
          *
0000 F900        B     BOOT
0001 0700
          *
          *      Begin Interrupt Routine
          *
0002 4000 INTRPT IN    INPUT,0
0003 587E        SACH  IACH          ) Store only the
0004 507F        SACL  IACL          ) accumulator.
0005 693E        DMOV  BPF1
```

```
                *
                *       Begin 31-tap FIR BPF
                *
                *       Only 15 multiplies are
                *       needed because 16 of
                *       the coefficients are
                *       coded as zeros.
                *
0006 7F89               ZAC
0007 6A1D               LT      Z29
0008 6D2D               MPY     TAP30
0009 691C               DMOV    Z28
000A 681B               LTD     Z27
000B 6D2C               MPY     TAP28
000C 691A               DMOV    Z26
000D 6819               LTD     Z25
000E 6D2B               MPY     TAP26
000F 6918               DMOV    Z24
0010 6817               LTD     Z23
0011 6D2A               MPY     TAP24
0012 6916               DMOV    Z22
0013 6815               LTD     Z21
0014 6D29               MPY     TAP22
0015 6914               DMOV    Z20
0016 6813               LTD     Z19
0017 6D28               MPY     TAP20
0018 6912               DMOV    Z18
0019 6811               LTD     Z17
001A 6D27               MPY     TAP18
001B 6910               DMOV    Z16
001C 680F               LTD     Z15
001D 6D26               MPY     TAP16
001E 690E               DMOV    Z14
001F 680D               LTD     Z13
0020 6D25               MPY     TAP14
0021 690C               DMOV    Z12
0022 680B               LTD     Z11
0023 6D24               MPY     TAP12
0024 690A               DMOV    Z10
0025 6809               LTD     Z9
0026 6D23               MPY     TAP10
0027 6908               DMOV    Z8
0028 6807               LTD     Z7
0029 6D22               MPY     TAP8
002A 6906               DMOV    Z6
002B 6805               LTD     Z5
002C 6D21               MPY     TAP6
002D 6904               DMOV    Z4
002E 6803               LTD     Z3
002F 6D20               MPY     TAP4
0030 6902               DMOV    Z2
0031 6801               LTD     Z1
0032 6D1F               MPY     TAP2
0033 6800               LTD     INPUT
0034 0F45               ADD     ONE,15          } Add for roundoff.
0035 583E               SACH    BPF1,0
                *
```

```
                   *       End of BPF Code
                   *
0036
0036 6880  CONT    LARP    0
0037 F400          BANZ    RETURN
0038 0043
0039 7003          LARK    0,3
003A 6945          DMOV    ONE                ) Set OUTFLG.
                   *
003B 6881          LARP    1
003C F400          BANZ    RETURN
003D 0043
003E 7104          LARK    1,4
003F 693F          DMOV    BPF2
0040 203E          LAC     BPF1
0041 5041          SACL    PILOT1
0042 6946          DMOV    OUTFLG             ) Set SNCFLG.
                   *
                   *       Restore Accumulator for
                   *       Return from Interrupt
                   *
0043 657E  RETURN  ZALH    IACH
0044 617F          ADDS    IACL
0045 7F82          EINT
0046 7F8D          RET
                   *
                   *       Start of Non-Interrupt Code for
                   *       Modifying Delay Buffer Pointers
                   *       and Transmitting Data to Main
                   *       Processing Board.
                   *
0047 2046  WAIT    LAC     OUTFLG             ) Test for time to send.
0048 FF00          BZ      WAIT
0049 0047
004A 2048          LAC     DLYADX
004B 6742          TBLR    DBPF1              )
004C 703E          TBLW    BPF1               ) Read in delayed data
004D 0045          ADD     ONE                ) and read out present
004E 6743          TBLR    DBPF2              ) filter output.
004F 703F          TBLW    BPF2               )
0050 0045          ADD     ONE
0051 5048          SACL    DLYADX
                   *
0052 7F89          ZAC
0053 5046          SACL    OUTFLG             ) Clear flag.
0054 2047          LAC     SNCFLG             ) Test for time to send
0055 FF00          BZ      NOSYNC             ) pilot.
0056 005E
                   *
0057 4B47          OUT     SNCFLG,3           )
0058 4F42          OUT     DBPF1,7            ) Output pilot/data
0059 4E41          OUT     PILOT1,6           ) sync, data & pilot
005A 4D43          OUT     DBPF2,5            ) streams.
005B 4C40          OUT     PILOT2,4           )
005C F900          B       MODPTR
005D 0060
                   *
```

```
005E 4F42   NOSYNC OUT    DBPF1,7        } Output only data
005F 4D43          OUT    DBPF2,5        } streams.
                   *
0060 2048   MODPTR LAC    DLYADX         } Wrap delay buffer
0061 104A          SUB    FIFOND         } pointer if needed.
0062 FE00          BNZ    WAIT
0063 0047
                   *
0064 2049          LAC    DLYBEG
0065 5048          SACL   DLYADX
0066 F900          B      WAIT
0067 0047
                   *
                   *      Reset Code for Initialization
                   *      of Constants and Pointers
                   *
0068 7F81   RESET  DINT
0069 6E00          LDPK   0
006A 4000          IN     0,0
006B 7F8B          SOVM
006C 707F          LARK   0,>7F
006D 6880          LARP   0
006E 7F89          ZAC
006F 5088   CLRRAM SACL   *              } Zero internal
0070 F400          BANZ   CLRRAM         } ram.
0071 006F
0072 7E01          LACK   >1
0073 5045          SACL   ONE            }
0074 6A45          LT     ONE            }       .
0075 8340          MPYK   DLYSIZ         }       .
0076 7F8E          PAC                   } Store constants.
0077 504A          SACL   FIFOND         }       .
0078 8200          MPYK   BUFBEG         }       .
0079 7F8E          PAC                   }
007A 5049          SACL   DLYBEG         }
                   *
007B 8400          MPYK   COEFFS
007C 7F8E          PAC
007D 700E          LARK   0,14
007E 711F          LARK   1,>1F
007F 6881   LOAD   LARP   1              } Load BPF
0080 67A0          TBLR   *+,0           } coefficients.
0081 0045          ADD    ONE
0082 F400          BANZ   LOAD
0083 007F
                   *
0084 7003          LARK   0,3            } Initialize
0085 7104          LARK   1,4            } AR's.
                   *
0086 7F82          EINT
0087 F900          B      WAIT
0088 0047
                   *
                   *      Filter Coefficients for 48kHz BPF
                   *             Coded into 16 bits.
                   *
0400               AORG   >400
```

```
                   *
     0400 FFDE            DATA     -34,-775,1994,-1980,-1551,8822,-16544,19888
     0401 FCF9
     0402 07CA
     0403 F844
     0404 F9F1
     0405 2276
     0406 BF60
     0407 4DB0
     0408 BF60            DATA     -16544,8822,-1551,-1980,1994,-775,-34
     0409 2276
     040A F9F1
     040B F844
     040C 07CA
     040D FCF9
     040E FFDE
                   *
                   *      Boot Routine for Loading Program
                   *      Memory from EPROM to RAM
                   *
     0700                 AORG     >700
                   *
     0700 7E01     BOOT   LACK     >1
     0701 5000            SACL     >0
     0702 6A00            LT       >0
     0703 87FF            MPYK     >7FF
     0704 7F8E            PAC
     0705 670A     NOTDUN TBLR     >A
     0706 7D0A            TBLW     >A
     0707 1000            SUB      >0
     0708 FD00            BGEZ     NOTDUN
     0709 0705
     070A 8068            MPYK     RESET
     070B 7F8E            PAC
     070C 500A            SACL     >A
     070D 7E01            LACK     >1
     070E 7D0A            TBLW     >A
     070F F900            B        RESET
     0710 0068
                   *
                          END
NO ERRORS, NO WARNINGS
```

APPENDIX II

TMS320 MTCT PREMODULATION PROCESSING SOFTWARE

```
                          IDT     'MODUL'
                    *
     0000                  AORG    >0
                    *
     0000 F900            B       BOOT
     0001 020E
                    *
                    *   PROGRAM TO IMPLEMENT TCT TRANSMIT BASEBAND PROCESSING.
                    *   THIS CONTAINS THE TWO PATHS NEEDED TO
                    *   IMPLEMENT THE ENTIRE SYSTEM. DATA IS INPUT AS 0 OR 1,
                    *   IS THEN MANCHESTER ENCODED, RAISED COSINE PULSE SHAPED,
                    *   AND FINALLY HIGH PASS FILTERED. WE ASSUME HERE THAT
                    *   DATA IS BEING RECEIVED INTO TRANSMITTER AT 2.4 KBPS.
                    *   THE DATA IS SPLIT INTO ODD AND EVEN STREAM, AND THEN
                    *   MANCHESTER ENCODED, SO THE RATE STAYS AT 2.4 KBPS.
                    *   RAISED COSINE PULSE SHAPING IS DONE USING LAST EIGHT
                    *   CODE BITS, WITH 4 SAMPLES BEING OUTPUT FOR EACH CODE
                    *   BIT. THE OUTPUT IS THEREFORE CLOCKING OUT AT 9.6 KPBS,
                    *   AND A NEW INPUT IS TAKEN ONCE EVERY FOUR OUTPUTS.
                    *
                    *   INPUT COEFFICIENTS NEEDED FOR PULSE SHAPING, CLOCK, ETC.
                    *
     0010                 AORG    >10
                    *
     0010 0024      C1     DATA    36      * PULSE SHAPING COEFFICIENT P(1)
     0011 005B      C2     DATA    91      * P(2)  ALL PULSE SHAPING COEFFICIEN
     0012 004C      C3     DATA    76      * P(3)  ARE SCALED BY 16384
     0013 000E      C4     DATA    14      * P(4)
     0014 0013      C5     DATA    19      * P(5)
     0015 00AD      C6     DATA    173     * P(6)
     0016 016B      C7     DATA    363     * P(7)
     0017 0106      C8     DATA    262     * P(8)
     0018 FE61      C9     DATA    -415    * P(9)
     0019 FA21      C10    DATA    -1503   * P(10)
     001A F776      C11    DATA    -2186   * P(11)
     001B FAE9      C12    DATA    -1303   * P(12)
     001C 076B      C13    DATA    1899    * P(13)
     001D 1B74      C14    DATA    7028    * P(14)
     001E 308F      C15    DATA    12431   * P(15)
     001F 3E24      C16    DATA    15908   * P(16)
     0020 FFFF      C17    DATA    -1      * HOLDS -1 FOR MANCHESTER CODING
     0021 8000      C18    DATA    >8000   * BIAS TERM FOR INTERFACE TO BIPOLAR
     0022 000A      C19    DATA    10      * AIB BOARD CLOCK PARAMETER
     0023 0200      C20    DATA    512     * AIB BOARD CLOCK PARAMETER
                    *
                    *      WRITE CONSTANTS TO DATA MEMORY
                    *
                    *
                    *
     0024 7F80      RESET  NOP
     0025 7F80             NOP
     0026 7F81            DINT
     0027 7F83             SOVM
     0028 6E00             LDPK    0
     0029 7E10             LACK    C1      * P(1)
     002A 6700             TBLR    0
     002B 7E11             LACK    C2      * P(2)
```

```
002C 6701            TBLR    1
002D 7E12            LACK    C3       * P(3)
002E 6702            TBLR    2
002F 7E13            LACK    C4       * P(4)
0030 6703            TBLR    3
0031 7E14            LACK    C5       * P(5)
0032 6704            TBLR    4
0033 7E15            LACK    C6       * P(6)
0034 6705            TBLR    5
0035 7E16            LACK    C7       * P(7)
0036 6706            TBLR    6
0037 7E17            LACK    C8       * P(8)
0038 6707            TBLR    7
0039 7E18            LACK    C9       * P(9)
003A 6708            TBLR    8
003B 7E19            LACK    C10      * P(10)
003C 6709            TBLR    9
003D 7E1A            LACK    C11      * P(11)
003E 670A            TBLR    10
003F 7E1B            LACK    C12      * P(12)
0040 670B            TBLR    11
0041 7E1C            LACK    C13      * P(13)
0042 670C            TBLR    12
0043 7E1D            LACK    C14      * P(14)
0044 670D            TBLR    13
0045 7E1E            LACK    C15      * P(15)
0046 670E            TBLR    14
0047 7E1F            LACK    C16      * P(16)
0048 670F            TBLR    15
0049 6E01            LDPK    1
004A 7E20            LACK    C17      * -1
004B 6705            TBLR    5
004C 7E21            LACK    C18      * BIAS
004D 6706            TBLR    6
004E 7E22            LACK    C19      * CLOCK CONSTANT
004F 6708            TBLR    8
0050 7E23            LACK    C20      * CLOCK CONSTANT
0051 6709            TBLR    9
                *
0052 7E00            LACK    0
0053 5003            SACL    3
0054 5004            SACL    4
0055 F900              B         MAN
0056 0057
                *
                *   MAIN CODE LOOP
                *  MANCHESTER CODING SECTION
                *
                *     CODE ODD BIT.
                *
0057 6E01       MAN     LDPK    1
0058 2003               LAC     3       *LOAD ODD DATA BIT INTO ACCUMULATOR
0059 FF00               BZ      ZERO    *BRANCH TO ZERO SECTION IF ZERO
005A 0062
                *
                *  HERE, A 1 BECOMES -1,1
                *
```

```
005B 2005              LAC     5         *LOAD ACCUM. WITH -1
005C 6E00              LDPK    0
005D 5070              SACL    112       *STORE FIRST MANCHESTER BIT,OMB(N) I
               *                          HERE, IT WILL BE READY FOR PULSE SH
005E 7E01              LACK    1         *LOAD ACCUM. WITH 1, SECOND MANCHEST
005F 6E01              LDPK    1
               *
0060 F900              B       DONE
0061 0067
               *
               *  HERE, A 0 BECOMES 1,-1
               *
0062 7E01      ZERO    LACK    1         *LOAD ACCUM. WITH 1
0063 6E00              LDPK    0
0064 5070              SACL    112       *STORE OMB(N) IN 112, WHERE IT WILL
0065 6E01              LDPK    1         *BE USED IMMEDIATELY FOR PULSE SHAPI
0066 2005              LAC     5         *LOAD ACCUM. WITH -1, SECOND MANCHES
               *
0067 5000      DONE    SACL    0         *STORE SECOND MAN. BIT, OMB(N+1) IN
               *                          THIS WILL BE USED IN NEXT PULSE SHA
               *
               *  NOW CODE EVEN BIT
               *
0068 2004              LAC     4         *LOAD DATA BIT INTO ACCUMULATOR
0069 FF00              BZ      ZERO1     *BRANCH TO ZERO SECTION IF ZERO
006A 0072
               *
               *  HERE,A 1 BECOMES -1,1
               *
006B 2005              LAC     5         *LOAD ACCUM. WITH -1
006C 6E00              LDPK    0         *STORE FIRST MANCHESTER BIT,EMB(N) I
006D 5078              SACL    120       *WHERE IT WILL BE USED IMMEDIATELY F
006E 6E01              LDPK    1         *PULSE SHAPING.
               *
006F 7E01              LACK    1         *LOAD ACCUM. WITH 1, SECOND MANCHEST
               *
0070 F900              B       DONE1
0071 0077
               *
               *  HERE, DATA BIT 0 BECOMES 1,-1
               *
0072 7E01      ZERO1   LACK    1         *LOAD ACCUM. WITH 1
0073 6E00              LDPK    0
0074 5078              SACL    120       *STORE EMB(N) IN 120 FOR PULSE SHAPI
0075 6E01              LDPK    1
0076 2005              LAC     5         *LOAD ACCUM. WITH -1, SECOND MANCHES
               *
0077 5001      DONE1   SACL    1         *STORE SECOND MAN. BIT, EMB(N+1) IN
               *                          THIS WILL BE STORED FOR ONE CYCLE.
               *
0078 7E01              LACK    1         *INITIALIZE MAN. BIT COUNTER (MCOUNT
0079 5007              SACL    7         *COUNTER STORE IN 135.
               *
               * FIND FIRST OF FOUR ODD OUTPUTS CORRESPONDING TO ONE MANCHE
               * PULSE SHAPING IS DONE CHRONOLOGICALLY. THE LAST EIGHT ODD
               * OMB(N-7) TO OMB(N), ARE STORED IN DMA'S 112-119, WITH 112
               * THE MOST RECENT.
```

```
                  *
                  *
007A 6E00  MAIN1     LDPK     0
                  *
007B 7F89            ZAC
007C 6A77            LT       119
007D 6D03            MPY      3          * OMB(N-7)*P(4)
                  *
007E 6C76            LTA      118
007F 6D07            MPY      7          * + OMB(N-6)*P(8)
                  *
0080 6C75            LTA      117
0081 6D0B            MPY      11         * + OMB(N-5)*P(12)
                  *
0082 6C74            LTA      116
0083 6D0F            MPY      15         * + OMB(N-4)*P(16)
                  *
0084 6C73            LTA      115
0085 6D0C            MPY      12         * + OMB(N-3)*P(13)
                  *
0086 6C72            LTA      114
0087 6D08            MPY      8          * + OMB(N-2)*P(9)
                  *
0088 6C71            LTA      113
0089 6D04            MPY      4          * + OMB(N-1)*P(5)
                  *
008A 6C70            LTA      112
008B 6D00            MPY      0          * + OMB(N)*P(1)
008C 7F8F            APAC
                  *
008D 5010            SACL     16         * STORE RESULT, ORC(M), IN DMA 16
008E 7007            LARK     0,7        * STORE FLAG IN AUX0, TO KEEP TRACK
                  *                        OUTPUT WE'RE ON.
008F 713C            LARK     1,60       * INITIALIZE POINTER AUX1 TO OLDEST
                  *                        IN ODD FILTER BUFFER FOR HI PASS F
                  *                        WHICH IS THE NEXT STEP.
0090 6881            LARP     1          * AUX1 POINTER PUT IN USE NOW.
                  *
0091 F900            B        FILTER     * BRANCH TO HIGH PASS FILTER SECTION
0092 0195
                  *
                  *   NOW DO FIRST EVEN BIT. SAME PULSE SHAPING AS DONE FOR
                  *   PREVIOUS ODD BIT. MANCHESTER BIT BUFFER IS IN DMA'S
                  *   120-127, WITH 120 HOLDING MOST RECENT BIT.
                  *
0093 690C  SEVEN     DMOV     12         *MOVE ODD OUTPUTS THROUGH DELAY BUFF
0094 690B            DMOV     11         * THIS BUFFER DELAYS ODD OUTPUT BY
0095 690A            DMOV     10         * TWO CLOCK CYCLES.
                  *  NOP
0096 6E00            LDPK     0
0097 7F89            ZAC
0098 6A7F            LT       127
0099 6D03            MPY      3          * EMB(N-7)*P(4)
                  *
009A 6C7E            LTA      126
009B 6D07            MPY      7          * + EMB(N-6)*P(8)
                  *
```

```
009C 6C7D            LTA     125
009D 600B            MPY     11       * + EMB(N-5)*P(12)
                *
009E 6C7C            LTA     124
009F 600F            MPY     15       * + EMB(N-4)*P(16)
                *
00A0 6C7B            LTA     123
00A1 600C            MPY     12       * + EMB(N-3)*P(13)
                *
00A2 6C7A            LTA     122
00A3 6008            MPY     8        * + EMB(N-2)*P(9)
                *
00A4 6C79            LTA     121
00A5 6004            MPY     4        * + EMB(N-1)*P(5)
                *
00A6 6C78            LTA     120
00A7 6000            MPY     0        * + EMB(N)*P(1)
00A8 7F8F            APAC
                *
00A9 503D            SACL    61       *STORE RESULT, ERC(N) IN 61
00AA 7006            LARK    0,6      *SET FLAG AUX0 TO MARK OUTPUT WE"RE
00AB 7169            LARK    1,105    *SET POINTER TO LAST ENTRY IN FILTER
00AC 6881            LARP    1        *CHOOSE POINTER.
                *
00AD F900            B       FILTER   *BRANCH TO HI PASS FILTER SECTION.
00AE 0195
                *
                *    SECOND OUTPUT OF FOUR. (ODD)
                *
00AF F600  SIX       BIOZ    LOOP2    * WAIT UNTIL READY FOR OUTPUT
00B0 00B3
00B1 F900            B       SIX
00B2 00AF
                *
00B3 4A0D  LOOP2     OUT     13,2     *OUTPUT DELAYED ODD SAMPLE TO PORT 2
00B4 480A            OUT     10,3     *OUTPUT EVEN SAMPLE TO PORT 3.
                *
00B5 6E00            LDPK    0
00B6 7F89            ZAC
00B7 6A77            LT      119
00B8 6002            MPY     2        * OMB(N-7)*P(3)
                *
00B9 6C76            LTA     118
00BA 6006            MPY     6        * + OMB(N-6)*P(7)
                *
00BB 6C75            LTA     117
00BC 600A            MPY     10       * + OMB(N-5)*P(11)
                *
00BD 6C74            LTA     116
00BE 600E            MPY     14       * + OMB(N-4)*P(15)
                *
00BF 6C73            LTA     115
00C0 600D            MPY     13       * + OMB(N-3)*P(14)
                *
00C1 6C72            LTA     114
00C2 6009            MPY     9        * + OMB(N-2)*P(10)
                *
```

```
00C3 6C71          LTA     113
00C4 6D05          MPY     5         * + OMB(N-1)*P(6)
             *
00C5 6C70          LTA     112
00C6 6D01          MPY     1         * + OMB(N)*P(2)
00C7 7F8F          APAC
             *
00C8 5010          SACL    16        * STORE RESULT ORC(M) IN DMA 16
00C9 7005          LARK    0,5       *SET FLAG TO MARK OUTPUT.
00CA 713C          LARK    1,60      *SET POINTER TO LAST SPOT IN FILTER
00CB 6881          LARP    1         *CHOOSE POINTER.
             *
00CC F900          B       FILTER    * BRANCH TO HIGH PASS FILTER.
00CD 0195
             *
             *  SECOND OUTPUT - EVEN BIT
             *
00CE 690C    FIVE  DMOV    12        *SHIFT ODD OUTPUTS THRU DELAY BUFFER
00CF 690B          DMOV    11
00D0 690A          DMOV    10
00D1 6E00          LDPK    0
00D2 7F89          ZAC
00D3 6A7F          LT      127
00D4 6D02          MPY     2         * EMB(N-7)*P(3)
             *
00D5 6C7E          LTA     126
00D6 6D06          MPY     6         * + EMB(N-6)*P(7)
             *
00D7 6C7D          LTA     125
00D8 6D0A          MPY     10        * + EMB(N-5)*P(11)
             *
00D9 6C7C          LTA     124
00DA 6D0E          MPY     14        * + EMB(N-4)*P(15)
             *
00DB 6C7B          LTA     123
00DC 6D0D          MPY     13        * + EMB(N-3)*P(14)
             *
00DD 6C7A          LTA     122
00DE 6D09          MPY     9         * + EMB(N-2)*P(10)
             *
00DF 6C79          LTA     121
00E0 6D05          MPY     5         * + EMB(N-1)*P(6)
             *
00E1 6C78          LTA     120
00E2 6D01          MPY     1         * + EMB(N)*P(2)
00E3 7F8F          APAC
             *
00E4 503D          SACL    61        *STORE EVEN PULSE SAMPLE IN DMA 61
00E5 7004          LARK    0,4       *SET FLAG SO WE KNOW WHERE WE ARE IN
00E6 7169          LARK    1,105     *SET POINTER TO LAST VALUES IN FILTE
00E7 6881          LARP    1         *SELECT POINTER.
             *
00E8 F900          B       FILTER    *BRANCH TO HI PASS FILTER.
00E9 0195
             *
             *
             *         THIRD OUTPUT OF FOUR ODD OUTPUTS
```

```
0331                *
0332 00EA F600  FOUR     BIOZ    LOOP3    * WAIT UNTIL READY FOR OUTPUT
     00EB 00EE
0333 00EC F900           B       FOUR
     00ED 00EA
0334                *
0335 00EE 4A0D  LOOP3    OUT     13,2     *OUTPUT DELAYED ODD SAMPLE TO PORT 2
0336 00EF 4B0A           OUT     10,3     *OUTPUT EVEN SAMPLE TO PORT 3.
0337                *
0338 00F0 6E00           LDPK    0
0339 00F1 7F89           ZAC
0340 00F2 6A77           LT      119
0341 00F3 6D01           MPY     1        * OMB(N-7)*P(2)
0342                *
0343 00F4 6C76           LTA     118
0344 00F5 6D05           MPY     5        * + OMB(N-6)*P(6)
0345                *
0346 00F6 6C75           LTA     117
0347 00F7 6D09           MPY     9        * + OMB(N-5)*P(10)
0348                *
0349 00F8 6C74           LTA     116
0350 00F9 6D0D           MPY     13       * + OMB(N-4)*P(14)
0351                *
0352 00FA 6C73           LTA     115
0353 00FB 6D0E           MPY     14       * + OMB(N-3)*P(15)
0354                *
0355 00FC 6C72           LTA     114
0356 00FD 6D0A           MPY     10       * + OMB(N-2)*P(11)
0357                *
0358 00FE 6C71           LTA     113
0359 00FF 6D06           MPY     6        * + OMB(N-1)*P(7)
0360                *
0361 0100 6C70           LTA     112
0362 0101 6D02           MPY     2        * + OMB(N)*P(3)
0363 0102 7F8F           APAC
0364                *
0365 0103 5010           SACL    16       * STORE RESULT ORC(M) IN DMA 16
0366 0104 7003           LARK    0,3      *SET FLAG TO MARK WHERE WE ARE IN PU
0367 0105 713C           LARK    1,60     *SET POINTER TO LAST VALUE IN FILTER
0368 0106 6881           LARP    1        *SELECT POINTER.
0369                *
0370 0107 F900           B       FILTER   * BRANCH TO HIGH PASS FILTER
     0108 0195
0371                *
0372                *  THIRD OUTPUT OUT OF FOUR EVEN OUTPUTS
0373                *
0374 0109 690C  THIRD    DMOV    12       *SHIFT ODD OUTPUTS THRU DELAY BUFFER
0375 010A 690B           DMOV    11
0376 010B 690A           DMOV    10
0377 010C 6E00           LDPK    0
0378 010D 7F89           ZAC
0379 010E 6A7F           LT      127
0380 010F 6D01           MPY     1        * EMB(N-7)*P(2)
0381                *
0382 0110 6C7E           LTA     126
0383 0111 6D05           MPY     5        * + EMB(N-6)*P(6)
0384                *
```

```
0112 6C7D            LTA     125
0113 6D09            MPY     9        * + EMB(N-5)*P(10)
                *
0114 6C7C            LTA     124
0115 6D0D            MPY     13       * + EMB(N-4)*P(14)
                *
0116 6C7B            LTA     123
0117 6D0E            MPY     14       * + EMB(N-3)*P(15)
                *
0118 6C7A            LTA     122
0119 6D0A            MPY     10       * + EMB(N-2)*P(11)
                *
011A 6C79            LTA     121
011B 6D06            MPY     6        * + EMB(N-1)*P(7)
                *
011C 6C78            LTA     120
011D 6D02            MPY     2        * + EMB(N)*P(3)
011E 7F8F            APAC
                *
011F 503D            SACL    61       *STORE EVEN PULSE SAMPLE IN 61
0120 7002            LARK    0,2      *SET FLAG TO MARK WHERE WE ARE IN PU
0121 7169            LARK    1,105    *SET POINTER TO LAST VALUE IN FILTER
0122 6881            LARP    1        *SELECT POINTER.
                *
0123 F900            B       FILTER   *BRANCH TO HI PASS FILTER SECTION.
0124 0195
                *
                *
                *    FOURTH ODD OUTPUT OF FOUR.
                *
0125 F600  TWO       BIOZ    LOOP4    *WAIT UNTIL READY FOR OUTPUT
0126 0129
0127 F900            B       TWO
0128 0125
                *
0129 4A0D  LOOP4     OUT     13,2     *OUTPUT DELAYED ODD SAMPLE TO PORT 2
012A 480A           OUT     10,3     *OUTPUT EVEN SAMPLE TO PORT 3.
                *
012B 6E00            LDPK    0
012C 7F89            ZAC
012D 6A77            LT      119
012E 6D00            MPY     0        * OMB(N-7)*P(1)
                *
012F 6C76            LTA     118
0130 6D04            MPY     4        * + OMB(N-6)*P(5)
                *
0131 6C75            LTA     117
0132 6D08            MPY     8        * + OMB(N-5)*P(9)
                *
0133 6C74            LTA     116
0134 6D0C            MPY     12       * + OMB(N-4)*P(13)
                *
0135 6C73            LTA     115
0136 6D0F            MPY     15       * + OMB(N-3)*P(16)
                *
0137 6C72            LTA     114
0138 6D0B            MPY     11       * + OMB(N-2)*P(12)
```

```
                  *
     0139 6C71          LTA     113
     013A 6D07          MPY     7       * + OMB(N-1)*P(8)
                  *
     013B 6C70          LTA     112
     013C 6D03          MPY     3       * + OMB(N)*P(4)
     013D 7F8F          APAC
                  *
     013E 5010          SACL    16      * STORE RESULT ORC(M) IN DMA 16
     013F 7001          LARK    0,1     *SET FLAG TO MARK WHERE WE ARE IN PU
     0140 713C          LARK    1,60    *SET POINTER TO LAST VALUE IN FILTER
     0141 6881          LARP    1       *SELECT POINTER.
                  *
     0142 F900          B       FILTER  * BRANCH TO HIGH PASS FILTER.
     0143 0195
                  *
                  *  FOURTH EVEN OUTPUT OUT OF FOUR.
                  *
     0144 690C    ONE   DMOV    12      *SHIFT ODD OUTPUTS THRU DELAY BUFFER
     0145 690B          DMOV    11
     0146 690A          DMOV    10
     0147 6E00          LDPK    0
     0148 7F89          ZAC
     0149 6A7F          LT      127
     014A 6D00          MPY     0       * EMB(N-7)*P(1)
                  *
     014B 6C7E          LTA     126
     014C 6D04          MPY     4       * + EMB(N-6)*P(5)
                  *
     014D 6C7D          LTA     125
     014E 6D08          MPY     8       * + EMB(N-5)*P(9)
                  *
     014F 6C7C          LTA     124
     0150 6D0C          MPY     12      * + EMB(N-4)*P(13)
                  *
     0151 6C7B          LTA     123
     0152 6D0F          MPY     15      * + EMB(N-3)*P(6)
                  *
     0153 6C7A          LTA     122
     0154 6D0B          MPY     11      * + EMB(N-2)*P(12)
                  *
     0155 6C79          LTA     121
     0156 6D07          MPY     7       * + EMB(N-1)*P(8)
                  *
     0157 6C78          LTA     120
     0158 6D03          MPY     3       * + EMB(N)*P(4)
     0159 7F8F          APAC
                  *
     015A 503D          SACL    61      *STORE EVEN SAMPLE ERC(N) IN DMA 61
     015B 7000          LARK    0,0     *SET FLAG TO MARK WHERE WE ARE IN PU
     015C 7169          LARK    1,105   *SET POINTER TO LAST VALUE IN FILTER
     015D 6881          LARP    1       *SELECT POINTER.
                  *
     015E F900          B       FILTER  *BRANCH TO HI PASS FILTER.
     015F 0195
                  *
                  *
```

```
                * NOW, MOVE INPUT BUFFER TO PREPARE FOR NEXT INCOMING MAN. B
                *
0160 6E00  PREP     LDPK    0
0161 6976           DMOV    118      * MOVE FIRST SEVEN VALUES OF MANCHES
0162 6975           DMOV    117      * BIT BUFFERS UP ONE MEMORY LOCATION
0163 6974           DMOV    116      * MAKE ROOM FOR MOST RECENT CODE BIT
0164 6973           DMOV    115      * FOR NEXT PULSE SHAPING SEQUENCE.
0165 6972           DMOV    114
0166 6971           DMOV    113
0167 6970           DMOV    112
0168 697E           DMOV    126
0169 697D           DMOV    125
016A 697C           DMOV    124
016B 697B           DMOV    123
016C 697A           DMOV    122
016D 6979           DMOV    121
016E 6978           DMOV    120
           *
           *
           *  WE ARE READY FOR NEXT ITERATION OF FOUR OUTPUTS CORRESPON
           *  TO NEXT MANCHESTER BIT. CHECK COUNTER TO SEE IF WE NEED N
           *  DATA BIT.
           *
016F 6E01           LDPK    1
0170 2007           LAC     7        *CHECK MCOUNT. ADD -1 TO COUNTER.
0171 0005           ADD     5        * STORE RESULT AS MCOUNT
0172 5007           SACL    7
           *
           *  IF WE HAVE LOOPED ONCE, WE NEED TO LOAD SECOND MAN. BIT.
           *  IF WE HAVE LOOPED TWICE, WE NEED TO INPUT A NEW DATA BIT.
           *
0173 FF00           BZ      LOAD     *IF MCOUNT IS NOW ZERO, WE HAVE MOST
0174 0181
           *                          RECENT MAN. BITS ALREADY,
           *                          SO SKIP FOLLOWING INPUT SEQUENCE
           *
0175 F600  WAIT5    BIOZ    LOOP5    * WAIT FOR CLOCK
0176 0179
0177 F900           B       WAIT5
0178 0175
           *
0179 4A0D  LOOP5    OUT     13,2     * OUTPUT DELAYED ODD SAMPLE TO PORT 2
017A 4B0A           OUT     10,3     * OUTPUT EVEN SAMPLE TO PORT 3.
           *
017B 4104           IN      4,1      * INPUT NEXT EVEN DATA BIT FROM PORT
017C 7E01           LACK    1
017D 7904           AND     4        * AND WITH +1 TO OBTAIN 0 OR 1 FOR D
017E 5004           SACL    4        * STORE DATA BIT IN DMA 132.
           *
017F F900           B       MAN      * BRANCH TO MANCHESTER CODING SECTIO
0180 0057
           *
           *  IF ZERO, THEN LOAD NEXT MANCHESTER BITS INTO BUFFER,
           *  INPUT NEXT ODD DATA BIT,
           *  AND LOOP BACK TO BEGIN PULSE SHAPING AGAIN.
           *
0181 2000  LOAD     LAC     0        * MOVE DMB(N+1) TO DMA 112
```

```
0182 6E00           LDPK    0       * WHERE IT WILL BE READY FOR
0183 5070           SACL    112     * NEXT PASS THRU PULSE SHAPING.
0184 6E01           LDPK    1
0185 2001           LAC     1       * MOVE EMB(N+1) TO DMA 120
0186 6E00           LDPK    0       * WHERE IT WILL BE READY FOR
0187 5078           SACL    120     * NEXT PASS THRU PULSE SHAPING.
0188 6E01           LDPK    1
          *
0189 F600 WAIT6     BIOZ    LOOP6   * WAIT FOR CLOCK
018A 018D
018B F900           B       WAIT6
018C 0189
          *
018D 4A0D LOOP6     OUT     13,2   * OUTPUT DELAYED ODD SAMPLE TO PORT 2
018E 4B0A          OUT     10,3   * OUTPUT EVEN SAMPLE TO PORT 3
          *
018F 4103           IN      3,1     *INPUT FROM PORT 1.
0190 7E01           LACK    1
0191 7903           AND     3       *AND WITH 1 TO PRODUCE A 1 OR A 0.
0192 5003           SACL    3       *STORE RESULTING DATA BIT IN DMA 131
          *
0193 F900           B       MAIN1   * BRANCH TO BEGINNING OF PULSE SHAPI
0194 007A
          *
          *         THIS SECTION CONTAINS CODE FOR HI PASS FILTER.
          *         PREVIOUS 45 FUNCTION VALUES, RC(M) - RC(M-45) IN
          *         MEMORY LOCATIONS 16-60 FOR THE ODD STREAM AND IN 61-
          *         FOR THE EVEN STREAM. THE OLDEST VALUE IS POINTED TO
          *         THIS POINT BY AUX1. THE COEFFICIENTS ARE
          *         SD CONSTANTS, SCALED BY 4096.
          *
          *         NOW START FIR FILTER
          *
0195 7F89 FILTER    ZAC
0196 6A98           LT      *-
0197 9F8D           MPYK    -115    * RC(M-44)*-115
          *
0198 6B98           LTD     *-
0199 9FDE           MPYK    -34     * + RC(M-43)*-34
          *
019A 6B98           LTD     *-
019B 9FD9           MPYK    -39     * + RC(M-42)*-39
          *
019C 6B98           LTD     *-
019D 9FD5           MPYK    -43     * + RC(M-41)*-43
          *
019E 6B98           LTD     *-
019F 9FD0           MPYK    -48     * + RC(M-40)*-48
          *
01A0 6B98           LTD     *-
01A1 9FCB           MPYK    -53     * + RC(M-39)*-53
          *
01A2 6B98           LTD     *-
01A3 9FC6           MPYK    -58     * + RC(M-38)*-58
          *
01A4 6B98           LTD     *-
01A5 9FC1           MPYK    -63     * + RC(M-37)*-63
```

```
                *
01A6 6898           LTD     *-
01A7 9FBC           MPYK    -68      * + RC(M-36)*-68
                *
01A8 6898           LTD     *-
01A9 9FB7           MPYK    -73      * + RC(M-35)*-73
                *
01AA 6898           LTD     *-
01AB 9FB2           MPYK    -78      * + RC(M-34)*-78
                *
01AC 6898           LTD     *-
01AD 9FAD           MPYK    -83      * + RC(M-33)*-83
                *
01AE 6898           LTD     *-
01AF 9FA9           MPYK    -87      * + RC(M-32)*-87
                *
01B0 6898           LTD     *-
01B1 9FA4           MPYK    -92      * + RC(M-31)*-92
                *
01B2 6898           LTD     *-
01B3 9FA1           MPYK    -95      * + RC(M-30)*-95
                *
01B4 6898           LTD     *-
01B5 9F9D           MPYK    -99      * + RC(M-29)*-99
                *
01B6 6898           LTD     *-
01B7 9F9A           MPYK    -102     * + RC(M-28)*-102
                *
01B8 6898           LTD     *-
01B9 9F98           MPYK    -104     * + RC(M-27)*-104
                *
01BA 6898           LTD     *-
01BB 9F95           MPYK    -107     * + RC(M-26)*-107
                *
01BC 6898           LTD     *-
01BD 9F93           MPYK    -109     * + RC(M-25)*-109
                *
01BE 6898           LTD     *-
01BF 9F92           MPYK    -110     * + RC(M-24)*-110
                *
01C0 6898           LTD     *-
01C1 9F91           MPYK    -111     * + RC(M-23)*-111
                *
01C2 6898           LTD     *-
01C3 8F91           MPYK    3985     * + RC(M-22)*3985
                *
01C4 6898           LTD     *-
01C5 9F91           MPYK    -111     * + RC(M-21)*-111
                *
01C6 6898           LTD     *-
01C7 9F92           MPYK    -110     * + RC(M-20)*-110
                *
01C8 6898           LTD     *-
01C9 9F93           MPYK    -109     * + RC(M-19)*-109
                *
01CA 6898           LTD     *-
01CB 9F95           MPYK    -107     * + RC(M-18)*-107
```

```
                 *
01CC 6B98               LTD     *-
01CD 9F98               MPYK    -104    * + RC(M-17)*-104
                 *
01CE 6B98               LTD     *-
01CF 9F9A               MPYK    -102    * + RC(M-16)*-102
                 *
01D0 6B98               LTD     *-
01D1 9F9D               MPYK    -99     * + RC(M-15)*-99
                 *
01D2 6B98               LTD     *-
01D3 9FA1               MPYK    -95     * + RC(M-14)*-95
                 *
01D4 6B98               LTD     *-
01D5 9FA4               MPYK    -92     * + RC(M-13)*-92
                 *
01D6 6B98               LTD     *-
01D7 9FA9               MPYK    -87     * + RC(M-12)*-87
                 *
01D8 6B98               LTD     *-
01D9 9FAD               MPYK    -83     * + RC(M-11)*-83
                 *
01DA 6B98               LTD     *-
01DB 9FB2               MPYK    -78     * + RC(M-10)*-78
                 *
01DC 6B98               LTD     *-
01DD 9FB7               MPYK    -73     * + RC(M-9)*-73
                 *
01DE 6B98               LTD     *-
01DF 9FBC               MPYK    -68     * + RC(M-8)*-68
                 *
01E0 6B98               LTD     *-
01E1 9FC1               MPYK    -63     * + RC(M-7)*-63
                 *
01E2 6B98               LTD     *-
01E3 9FC6               MPYK    -58     * + RC(M-6)*-58
                 *
01E4 6B98               LTD     *-
01E5 9FCB               MPYK    -53     * + RC(M-5)*-53
                 *
01E6 6B98               LTD     *-
01E7 9FD0               MPYK    -48     * + RC(M-4)*-48
                 *
01E8 6B98               LTD     *-
01E9 9FD5               MPYK    -43     * + RC(M-3)*-43
                 *
01EA 6B98               LTD     *-
01EB 9FD9               MPYK    -39     * + RC(M-2)*-39
                 *
01EC 6B98               LTD     *-
01ED 9FDE               MPYK    -34     * + RC(M-1)*-34
                 *
01EE 6B98               LTD     *-
01EF 9F8D               MPYK    -115    * + RC(M)*-115
01F0 7F8F               APAC
                 *
01F1 6E01               LDPK    1
```

```
01F2 5C0A           SACH    10,4     *STORE FILTER OUTPUT IN 138
               *                     SHIFTED 4 SPOTS TO TAKE INTO ACCOUNT
               *                     MULT. OF 13 BIT NO. BY 16 BIT NO.
01F3 200A           LAC     10
01F4 0006           ADD     6        * ADD BIAS TERM TO READY FOR OUTPUT
               *     AND     15
01F5 500A           SACL    10       * STORE FILTER OUT + BIAS IN 138
               *
               *    CHECK OCOUNT TO SEE WHICH OUTPUT WE SHOULD BRANCH BACK
               *
01F6 300E           SAR     0,14
01F7 200E           LAC     14
               *
               * IF ACC. IS ZERO, WE ARE DONE AND READY FOR NEXT ITERATION
               *
01F8 FF00           BZ      PREP
01F9 0160
               *
01FA 0005           ADD     5
               *
               *   IF ZERO NOW, READY FOR FOURTH EVEN OUTPUT
               *
01FB FF00           BZ      ONE
01FC 0144
               *
01FD 0005           ADD     5
               *
               *  IF ZERO, READY FOR FOURTH ODD OUTPUT
               *
01FE FF00           BZ      TWO
01FF 0125
               *
0200 0005           ADD     5
               *
               *  IF ZERO, READY FOR THIRD EVEN OUTPUT
               *
0201 FF00           BZ      THIRD
0202 0109
               *
0203 0005           ADD     5
               *
               *  IF ZERO, READY FOR THIRD ODD OUTPUT
               *
0204 FF00           BZ      FOUR
0205 00EA
               *
0206 0005           ADD     5
               *
               *  IF ZERO, READY FOR SECOND EVEN OUTPUT
               *
0207 FF00           BZ      FIVE
0208 00CE
               *
0209 0005           ADD     5
               *
               *  IF ZERO, READY FOR SECOND ODD OUTPUT
               *
```

```
0766 020A FF00          BZ      SIX
     020B 00AF
0767               *
0768               *  IF ONE, READY FOR FIRST EVEN OUTPUT
0769               *
0770 020C FC00          BGZ     SEVEN
     020D 0093
0771               *
0772               *
0773               *  BOOT ROUTINE FOR LOADING PROGRAM
0774               *   MEMORY FROM EPROM TO RAM
0775               *
0776 020E 7E01     BOOT    LACK    >1
0777 020F 5000            SACL    >0
0778 0210 6A00            LT      >0
0779 0211 8700            MPYK    >700
0780 0212 7F8E            PAC
0781               *
0782 0213 670A     NOTDUN  TBLR    >A
0783 0214 7D0A            TBLW    >A
0784 0215 1000            SUB     >0
0785 0216 F000            BGEZ    NOTDUN
     0217 0213
0786               *
0787 0218 8024            MPYK    RESET
0788 0219 7F8E            PAC
0789 021A 500A            SACL    >A
0790 021B 7E01            LACK    >1
0791 021C 7D0A            TBLW    >A
0792 021D F900            B       RESET
     021E 0024
0793               *
0794                      END
0 ERRORS, NO WARNINGS
```

APPENDIX III

TMS320 MTCT DEMODULATOR PROCESSING SOFTWARE

```
                         IDT     'DEMPROC'
                *****************************************************
                *                                                   *
                *       SOFTWARE FOR BOARD 2 OF THE REAL TIME       *
                *       IMPLEMENTATION OF THE MANCHESTER TCT        *
                *                 DEMODULATOR                       *
                *                                                   *
                *       Written by - Norman E. Lay                  *
                *       Last Updated : 8/30/85                      *
                *                                                   *
                *       General Electric Company                    *
                *       Corporate Research & Development            *
                *       Schenectady, N.Y.                           *
                *                                                   *
                *---------------------------------------------------*
                *                                                   *
                *       The following TMS-320 assembler code        *
                *       implements the pilot processing and         *
                *       correction to the data channels of phase    *
                *       irregularities caused by the fading chan-   *
                *       nel. The principal processing intensive     *
                *       functions implemented by this code consist  *
                *       of 4 lowpass filters -- 2 for pilot reco-   *
                *       very in each channel and 2 for filtering    *
                *       excess noise from the data band. The pilot  *
                *       processor consists of determining sine &    *
                *       cosine of the phase angle between the I & Q *
                *       channels of the recovered pilot. One octant *
                *       of sine and cosine values are stored as a   *
                *       lookup table in program memory. Linear in-  *
                *       terpolation is used at different processing *
                *       rate boundaries (i.e. where the pilot is    *
                *       removed from the data and where sin & cos   *
                *       are used to correct for phase errors in the *
                *       data). The data filters operate at a 12kHz  *
                *       rate and the pilot processing is done at a  *
                *       2.4kHz rate.                                *
                *                                                   *
                *---------------------------------------------------*
                *
                *       1st Data Page Ram
                *
          0000  ILPF    EQU     >0      )
          0001  IZ1     EQU     >1      ) Beginning of Ram for
          0002  IZ2     EQU     >2      ) Delay Storage of Filter
          0003  IZ3     EQU     >3      ) States of the I Channel
          0004  IZ4     EQU     >4      )      Data LPF
          0005  IZ5     EQU     >5      )        .
          0006  IZ6     EQU     >6      )        .
          0007  IZ7     EQU     >7      )        .
          0008  IZ8     EQU     >8      )        .
          0009  IZ9     EQU     >9      )        .
          000A  IZ10    EQU     >A      )        .
          000B  IZ11    EQU     >B      )        .
          000C  IZ12    EQU     >C      )        .
          000D  IZ13    EQU     >D      )        .
          000E  IZ14    EQU     >E      )        .
```

```
000F   IZ15   EQU   >F    )       .
0010   IZ16   EQU   >10   )       .
0011   IZ17   EQU   >11   )       .
0012   IZ18   EQU   >12   )       .
0013   IZ19   EQU   >13   )       .
0014   IZ20   EQU   >14   )       .
0015   IZ21   EQU   >15   )       .
0016   IZ22   EQU   >16   )       .
0017   IZ23   EQU   >17   )       .
0018   IZ24   EQU   >18   )       .
0019   IZ25   EQU   >19   )       .
001A   IZ26   EQU   >1A   )       .
001B   IZ27   EQU   >1B   )       .
001C   IZ28   EQU   >1C   )       .
001D   IZ29   EQU   >1D   )       .
001E   IZ30   EQU   >1E   )       .
001F   IZ31   EQU   >1F   )       .
0020   IZ32   EQU   >20   )       .
0021   IZ33   EQU   >21   )       .
0022   IZ34   EQU   >22   )       .
0023   IZ35   EQU   >23   )       .
0024   IZ36   EQU   >24   )       .
0025   IZ37   EQU   >25   )       .
0026   IZ38   EQU   >26   )       .
0027   IZ39   EQU   >27   ) End of I Data LPF
0028   IZ40   EQU   >28   )   Delay Storage
              *
0029   QLPF   EQU   >29   )
002A   QZ1    EQU   >2A   ) Beginning of Ram for
002B   QZ2    EQU   >2B   ) Delay Storage of Filter
002C   QZ3    EQU   >2C   ) States of the Q Channel
002D   QZ4    EQU   >2D   )      Data LPF
002E   QZ5    EQU   >2E   )       .
002F   QZ6    EQU   >2F   )       .
0030   QZ7    EQU   >30   )       .
0031   QZ8    EQU   >31   )       .
0032   QZ9    EQU   >32   )       .
0033   QZ10   EQU   >33   )       .
0034   QZ11   EQU   >34   )       .
0035   QZ12   EQU   >35   )       .
0036   QZ13   EQU   >36   )       .
0037   QZ14   EQU   >37   )       .
0038   QZ15   EQU   >38   )       .
0039   QZ16   EQU   >39   )       .
003A   QZ17   EQU   >3A   )       .
003B   QZ18   EQU   >3B   )       .
003C   QZ19   EQU   >3C   )       .
003D   QZ20   EQU   >3D   )       .
003E   QZ21   EQU   >3E   )       .
003F   QZ22   EQU   >3F   )       .
0040   QZ23   EQU   >40   )       .
0041   QZ24   EQU   >41   )       .
0042   QZ25   EQU   >42   )       .
0043   QZ26   EQU   >43   )       .
0044   QZ27   EQU   >44   )       .
0045   QZ28   EQU   >45   )       .
0046   QZ29   EQU   >46   )       .
```

```
0047   QZ30   EQU   >47      )       .
0048   QZ31   EQU   >48      )       .
0049   QZ32   EQU   >49      )       .
004A   QZ33   EQU   >4A      )       .
004B   QZ34   EQU   >4B      )       .
004C   QZ35   EQU   >4C      )       .
004D   QZ36   EQU   >4D      )       .
004E   QZ37   EQU   >4E      )       .
004F   QZ38   EQU   >4F      )       .
0050   QZ39   EQU   >50      ) End of Q Data LPF
0051   QZ40   EQU   >51      )   Delay Storage
       *
0052   IDATA  EQU   >52      ) Ram for storing I & Q
0053   QDATA  EQU   >53      ) input data
       *
0054   IBUFF  EQU   >54      ) Input data buffer to
0059   QBUFF  EQU   >59      ) permit pilot processing
       *
005E   SIN    EQU   >5E      ) Sine calculation
005F   PRESIN EQU   >5F        ) Save for sine lin. interp.
0060   OLDSIN EQU   >60      ) Use as sine in interrupt
0061   COS    EQU   >61        ) Cosine calculation
0062   PRECOS EQU   >62      ) Save for cosine lin. interp.
0063   OLDCOS EQU   >63        ) Use as cosine in interrupt
0064   SINSTP EQU   >64      ) Sine step size calc.
0065   OSSTP  EQU   >65        ) Use in intrpt. as step size
0066   COSSTP EQU   >66      ) Cosine step size calc.
0067   OCSTP  EQU   >67        ) Use in intrpt. as step size
       *
0068   IPILOT EQU   >68      ) Input I pilot
0069   OLDIP  EQU   >69        ) Save for I lin. interp.
006A   OIPIL  EQU   >6A      ) Use as I pilot in interrupt
006B   QPILOT EQU   >6B        ) Input Q pilot
006C   OLDQP  EQU   >6C      ) Save for Q pilot lin. interp.
006D   OQPIL  EQU   >6D        ) Use as Q pilot in intrpt.
006E   IPSTP  EQU   >6E      ) I pilot step size calc.
006F   OIPSTP EQU   >6F        ) Use in intrpt. as step size
0070   QPSTP  EQU   >70      ) Q pilot step size calc.
0071   OQPSTP EQU   >71        ) Use in intrpt. as step size
       *
0072   ONE    EQU   >72      ) Constant = 5
0073   SNCFLG EQU   >73        ) Flag for pilot/data alignment
0074   TEMP1  EQU   >74      ) Temp ram used in background
0075   TEMP2  EQU   >75      )          .
0076   SINMI  EQU   >76      ) Temp ram used in interrupt
0077   SINMC  EQU   >77      )          .
0078   IDOUT  EQU   >78      ) I data output
0079   QDOUT  EQU   >79      ) Q data output
007A   SNOFST EQU   >7A        ) Table offset for stored sine
007A   CSOFST EQU   >7A        ) and cosine values
007B   ISIGN  EQU   >7B      ) Temp ram used in background
007C   QSIGN  EQU   >7C      )          .
       *
       *      Ram for Saving Registers
       *      During an Interrupt
       *
007D   IACH   EQU   >7D      ) Save high accumulator
```

```
        007E    IACL    EQU     >7E     ) Save low accumulator
        007F    ITREG   EQU     >7F     ) Save T register
                *
                *       Address Constant for Sine
                *       and Cosine Lookup Table
                *
        0500    OFFSET  EQU     >500
                *
                *       Data LPF Constants
                *
                *       Only half the coefficients are coded because
                *       the filter is symmetrical.
                *
        FFDB    DLPF1   EQU     -37
        FFE7    DLPF2   EQU     -25
        006C    DLPF3   EQU     108
        FFFF    DLPF4   EQU     -1
        FFD4    DLPF5   EQU     -44
        FFA6    DLPF6   EQU     -90
        FFF8    DLPF7   EQU     -8
        0068    DLPF8   EQU     104
        0078    DLPF9   EQU     120
        FFE6    DLPF10  EQU     -26
        FF44    DLPF11  EQU     -188
        FF67    DLPF12  EQU     -153
        006B    DLPF13  EQU     107
        0144    DLPF14  EQU     324
        00B5    DLPF15  EQU     181
        FEDA    DLPF16  EQU     -294
        FD9E    DLPF17  EQU     -610
        FF38    DLPF18  EQU     -200
        03E9    DLPF19  EQU     1001
        092C    DLPF20  EQU     2348
        0B79    DLPF21  EQU     2937
                *
                *       Pilot LPF Constants
                *
                *       Only half the coefficients are coded because
                *       the filter is symmetrical.
                *
        FF34    PLPF1   EQU     -204
        FFE6    PLPF2   EQU     -26
        FFF6    PLPF3   EQU     -10
        0011    PLPF4   EQU     17
        0036    PLPF5   EQU     54
        0060    PLPF6   EQU     96
        008B    PLPF7   EQU     139
        00B0    PLPF8   EQU     176
        00C8    PLPF9   EQU     200
        00CB    PLPF10  EQU     203
        00B4    PLPF11  EQU     180
        0081    PLPF12  EQU     129
        0031    PLPF13  EQU     49
        FFC7    PLPF14  EQU     -57
        FF4D    PLPF15  EQU     -179
        FECD    PLPF16  EQU     -307
        FE57    PLPF17  EQU     -425
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
               FDFB   PLPF18 EQU      -517
               FDCA   PLPF19 EQU      -566
               FDD4   PLPF20 EQU      -556
               FE27   PLPF21 EQU      -473
               FECA   PLPF22 EQU      -310
               FFC1   PLPF23 EQU      -63
               0106   PLPF24 EQU      262
               028E   PLPF25 EQU      654
               0446   PLPF26 EQU      1094
               0617   PLPF27 EQU      1559
               07E5   PLPF28 EQU      2021
               0994   PLPF29 EQU      2452
               0B06   PLPF30 EQU      2822
               0C23   PLPF31 EQU      3107
               0CD6   PLPF32 EQU      3286
               0D13   PLPF33 EQU      3347
                      *
                      *      Begin TMS-320 Code
                      *
        0000                 AORG     >0
                      *
        0000 F900            B        BOOT
        0001 0700
                      *
                      *      Begin Interrupt Routine
                      *
        0002 587D     INTRPT SACH     IACH               }
        0003 507E            SACL     IACL               } Save accumulator
        0004 8001            MPYK     >1                 } and T register dur-
        0005 7F8E            PAC                         } ing an interrupt.
        0006 507F            SACL     ITREG              }
                      *
        0007 4873            OUT      IDOUT,0            } Output recovered I
        0008 4979            OUT      QDOUT,1            } and Q data.
        0009 F600            BIOZ     NCSYNC             } Check for time alignment.
        000A 0020
                      *
        000B 6961            DMOV     COS                } Update sin, cos, I pilot
        000C 6962            DMOV     PRECOS             } and Q pilot and all cor-
        000D 6966            DMOV     COSSTP             } responding step sizes.
        000E 695E            DMOV     SIN                }       .
        000F 695F            DMOV     PRESIN             }       .
        0010 6964            DMOV     SINSTP             }       .
        0011 6968            DMOV     IPILOT             }       .
        0012 6969            DMOV     OLDIP              }       .
        0013 696E            DMOV     IPSTP              }       .
        0014 6963            DMOV     QPILOT             }       .
        0015 696C            DMOV     OLDQP              }       .
        0016 6970            DMOV     QPSTP              } End of Update.
                      *
        0017 6972            DMOV     ONE                } Set SNCFLG.
                      *
        0018 4052            IN       IDATA,0            }
        0019 4153            IN       QDATA,1            } Input I & Q pilot and
        001A 4268            IN       IPILOT,2           } data streams.
        001B 436B            IN       QPILOT,3           }
                      *
```

```
001C 7054          LARK    0,IBUFF         } Reset processing delay
001D 7159          LARK    1,QBUFF         } buffer pointers.
            *
001E F900          B       CONTNU
001F 002E
            *
0020 4052   NOSYNC IN      IDATA,0         } Input I & Q data
0021 4153          IN      QDATA,1         } streams.
            *
0022 656A          ZALH    OIPIL
0023 606F          ADDH    OIPSTP          }
0024 586A          SACH    OIPIL           } Update filtered pilot
0025 656D          ZALH    OQPIL           } for linear interpolation.
0026 6071          ADDH    OQPSTP          }
0027 586D          SACH    OQPIL
0028 6563          ZALH    OLDCOS
0029 6067          ADDH    OCSTP           }
002A 5863          SACH    OLDCOS          } Update sin & cos for
002B 6560          ZALH    OLDSIN          } linear interpolation.
002C 6065          ADDH    OSSTP           }
002D 5860          SACH    OLDSIN
            *
            *      The following section of code
            *      implements the equations :
            *
            *      Zi = Id*cos(phi) + Qd*sin(phi)
            *      Zq = Qd*cos(phi) - Id*sin(phi)
            *
002E 2088   CONTNU LAC     *               }
002F 106A          SUB     OIPIL           } Remove pilot from I data.
0030 5088          SACL    *               }
0031 6A60          LT      OLDSIN
0032 6D81          MPY     *,1
0033 7F8E          PAC
0034 5976          SACH    SINMI,1
0035 2088          LAC     *               }
0036 106D          SUB     OQPIL           } Remove pilot from Q data.
0037 5088          SACL    *               }
0038 6D80          MPY     *,0
0039 7F8E          PAC
003A 5977          SACH    SINMQ,1
003B 6A63          LT      OLDCOS
003C 6D81          MPY     *,1
003D 7F8E          PAC
003E 0F77          ADD     SINMQ,15
003F 5800          SACH    ILPF,0
0040 6D80          MPY     *,0
0041 7F8E          PAC
0042 1F76          SUB     SINMI,15
0043 5829          SACH    QLPF,0
            *
0044 2052          LAC     IDATA
0045 50A1          SACL    *+,0,1          } Store Current Input
0046 2053          LAC     QDATA           } I & Q Data.
0047 50A0          SACL    *+,0,0
            *
            *      The following two LPFs are for
```

```
                  *       the removal of out of band noise
                  *       in both the I & Q data channels.
                  *
                  *
                  *       Data LPF Code I Channel
                  *
0048 7F89                 ZAC
0049 6A28                 LT      IZ40
004A 9FD8                 MPYK    DLPF1
004B 6B27                 LTD     IZ39
004C 9FE7                 MPYK    DLPF2
004D 6B26                 LTD     IZ38
004E 806C                 MPYK    DLPF3
004F 6B25                 LTD     IZ37
0050 9FFF                 MPYK    DLPF4
0051 6B24                 LTD     IZ36
0052 9FD4                 MPYK    DLPF5
0053 6B23                 LTD     IZ35
0054 9FA6                 MPYK    DLPF6
0055 6B22                 LTD     IZ34
0056 9FF8                 MPYK    DLPF7
0057 6B21                 LTD     IZ33
0058 8068                 MPYK    DLPF8
0059 6B20                 LTD     IZ32
005A 8078                 MPYK    DLPF9
005B 6B1F                 LTD     IZ31
005C 9FE6                 MPYK    DLPF10
005D 6B1E                 LTD     IZ30
005E 9F44                 MPYK    DLPF11
005F 6B1D                 LTD     IZ29
0060 9F67                 MPYK    DLPF12
0061 6B1C                 LTD     IZ28
0062 806B                 MPYK    DLPF13
0063 6B1B                 LTD     IZ27
0064 8144                 MPYK    DLPF14
0065 6B1A                 LTD     IZ26
0066 8085                 MPYK    DLPF15
0067 6B19                 LTD     IZ25
0068 9EDA                 MPYK    DLPF16
0069 6B18                 LTD     IZ24
006A 9D9E                 MPYK    DLPF17
006B 6B17                 LTD     IZ23
006C 9F38                 MPYK    DLPF18
006D 6B16                 LTD     IZ22
006E 83E9                 MPYK    DLPF19
006F 6B15                 LTD     IZ21
0070 892C                 MPYK    DLPF20
0071 6B14                 LTD     IZ20
0072 8379                 MPYK    DLPF21
0073 6B13                 LTD     IZ19
0074 892C                 MPYK    DLPF20
0075 6B12                 LTD     IZ18
0076 83E9                 MPYK    DLPF19
0077 6B11                 LTD     IZ17
0078 9F38                 MPYK    DLPF18
0079 6B10                 LTD     IZ16
007A 9D9E                 MPYK    DLPF17
```

```
007B 6B0F           LTD     IZ15
007C 9EDA           MPYK    DLPF16
007D 6B0E           LTD     IZ14
007E 8065           MPYK    DLPF15
007F 6B0D           LTD     IZ13
0080 8144           MPYK    DLPF14
0081 6B0C           LTD     IZ12
0082 806B           MPYK    DLPF13
0083 6B0B           LTD     IZ11
0084 9F67           MPYK    DLPF12
0085 6B0A           LTD     IZ10
0086 9F44           MPYK    DLPF11
0087 6B09           LTD     IZ9
0088 9FE6           MPYK    DLPF10
0089 6B08           LTD     IZ8
008A 8078           MPYK    DLPF9
008B 6B07           LTD     IZ7
008C 8068           MPYK    DLPF8
008D 6B06           LTD     IZ6
008E 9FF8           MPYK    DLPF7
008F 6B05           LTD     IZ5
0090 9FA6           MPYK    DLPF6
0091 6B04           LTD     IZ4
0092 9FD4           MPYK    DLPF5
0093 6B03           LTD     IZ3
0094 9FFF           MPYK    DLPF4
0095 6B02           LTD     IZ2
0096 806C           MPYK    DLPF3
0097 6B01           LTD     IZ1
0098 9FE7           MPYK    DLPF2
0099 6B00           LTD     ILPF
009A 9FD3           MPYK    DLPF1
009B 7F8F           APAC
009C 5C78           SACH    ICOUT,4
               *
               *    Data LPF Code Q Channel
               *
009D 7F89           ZAC
009E 6A51           LT      QZ40
009F 9FD3           MPYK    DLPF1
00A0 6B50           LTD     QZ39
00A1 9FE7           MPYK    DLPF2
00A2 6B4F           LTD     QZ38
00A3 806C           MPYK    DLPF3
00A4 6B4E           LTD     QZ37
00A5 9FFF           MPYK    DLPF4
00A6 6B4D           LTD     QZ36
00A7 9FD4           MPYK    DLPF5
00A8 6B4C           LTD     QZ35
00A9 9FA6           MPYK    DLPF6
00AA 6B4B           LTD     QZ34
00AB 9FF8           MPYK    DLPF7
00AC 6B4A           LTD     QZ33
00AD 8068           MPYK    DLPF8
00AE 6B49           LTD     QZ32
00AF 8078           MPYK    DLPF9
00B0 6B48           LTD     QZ31
```

```
0081 9FE6    MPYK   DLPF10
0082 6B47    LTD    QZ30
0083 9F44    MPYK   DLPF11
0084 6B46    LTD    QZ29
0085 9F67    MPYK   DLPF12
0086 6B45    LTD    QZ28
0087 806B    MPYK   DLPF13
0088 6B44    LTD    QZ27
0089 8144    MPYK   DLPF14
008A 6B43    LTD    QZ26
008B 80B5    MPYK   DLPF15
008C 6B42    LTD    QZ25
008D 9EDA    MPYK   DLPF16
008E 6B41    LTD    QZ24
008F 9D9E    MPYK   DLPF17
00C0 6B40    LTD    QZ23
00C1 9F38    MPYK   DLPF18
00C2 6B3F    LTD    QZ22
00C3 83E9    MPYK   DLPF19
00C4 6B3E    LTD    QZ21
00C5 892C    MPYK   DLPF20
00C6 6B3D    LTD    QZ20
00C7 8B79    MPYK   DLPF21
00C8 6B3C    LTD    QZ19
00C9 892C    MPYK   DLPF20
00CA 6B3B    LTD    QZ18
00CB 83E9    MPYK   DLPF19
00CC 6B3A    LTD    QZ17
00CD 9F38    MPYK   DLPF18
00CE 6B39    LTD    QZ16
00CF 9D9E    MPYK   DLPF17
00D0 6B38    LTD    QZ15
00D1 9EDA    MPYK   DLPF16
00D2 6B37    LTD    QZ14
00D3 80B5    MPYK   DLPF15
00D4 6B36    LTD    QZ13
00D5 8144    MPYK   DLPF14
00D6 6B35    LTD    QZ12
00D7 806B    MPYK   DLPF13
00D8 6B34    LTD    QZ11
00D9 9F67    MPYK   DLPF12
00DA 6B33    LTD    QZ10
00DB 9F44    MPYK   DLPF11
00DC 6B32    LTD    QZ9
00DD 9FE6    MPYK   DLPF10
00DE 6B31    LTD    QZ8
00DF 8078    MPYK   DLPF9
00E0 6B30    LTD    QZ7
00E1 8068    MPYK   DLPF8
00E2 6B2F    LTD    QZ6
00E3 9FF8    MPYK   DLPF7
00E4 6B2E    LTD    QZ5
00E5 9FA6    MPYK   DLPF6
00E6 6B2D    LTD    QZ4
00E7 9FD4    MPYK   DLPF5
00E8 6B2C    LTD    QZ3
00E9 9FFF    MPYK   DLPF4
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00EA 6828         LTD     QZ2
00EB 806C         MPYK    DLPF3
00EC 682A         LTD     QZ1
00ED 9FE7         MPYK    DLPF2
00EE 6829         LTD     QLPF
00EF 9FD5         MPYK    DLPF1
00F0 7F8F         APAC
00F1 5C79         SACH    QDOUT,4
           *
           *      Restore Registers
           *
00F2 657D         ZALH    IACH
00F3 617E         ADDS    IACL
00F4 6A7F         LT      ITREG
           *
00F5 7F82         EINT
00F6 7F8D         RET
           *
           *      Begin Background Code for
           *      Calculation of I & Q Pilot
           *      and Sin & Cos Step Sizes.
           *      Perform Pilot LPF and Sin &
           *      Cos Table Lookup.
           *
00F7 2073  BKGRND LAC     SNCFLG          } Wait for pilot/data
00F8 FF00         BZ      BKGRND          } synchronization.
00F9 00F7
           *
           *      The following two LPFs are for pilot
           *      recovery of the I & Q channel.
           *
           *      Note : the limited internal memory of the
           *             TMS-320 required that the pilot LPF
           *             filter states be stored externally;
           *             however, TBLW/TBLR instructions carry
           *             a 6 cycle overhead for memory access;
           *             to lower this overhead penalty, extra
           *             ram was added to be accessed through
           *             the I/O ports reducing the overhead
           *             to 4 cycles; memory addressing is ac-
           *             complished by an external counter that
           *             is advanced by an OUT ,7 instruction;
           *
           *      I Pilot LPF Code
           *
00FA 7F89         ZAC
00FB 6A68         LT      IPILOT
00FC 9F34         MPYK    PLPF1
00FD 4774         IN      TEMP1,7
00FE 4F68         OUT     IPILOT,7
00FF 6C74         LTA     TEMP1
0100 9FE6         MPYK    PLPF2
0101 4768         IN      IPILOT,7
0102 4F74         OUT     TEMP1,7
0103 6C68         LTA     IPILOT
0104 9FF6         MPYK    PLPF3
0105 4774         IN      TEMP1,7
```

```
0106 4F68          OUT     IPILOT,7
0107 6C74          LTA     TEMP1
0108 8011          MPYK    PLPF4
0109 4768          IN      IPILOT,7
010A 4F74          OUT     TEMP1,7
010B 6C68          LTA     IPILOT
010C 8036          MPYK    PLPF5
010D 4774          IN      TEMP1,7
010E 4F68          OUT     IPILOT,7
010F 6C74          LTA     TEMP1
0110 8060          MPYK    PLPF6
0111 4768          IN      IPILOT,7
0112 4F74          OUT     TEMP1,7
0113 6C68          LTA     IPILOT
0114 8083          MPYK    PLPF7
0115 4774          IN      TEMP1,7
0116 4F68          OUT     IPILOT,7
0117 6C74          LTA     TEMP1
0118 8080          MPYK    PLPF8
0119 4768          IN      IPILOT,7
011A 4F74          OUT     TEMP1,7
011B 6C68          LTA     IPILOT
011C 80C8          MPYK    PLPF9
011D 4774          IN      TEMP1,7
011E 4F68          OUT     IPILOT,7
011F 6C74          LTA     TEMP1
0120 80C8          MPYK    PLPF10
0121 4768          IN      IPILOT,7
0122 4F74          OUT     TEMP1,7
0123 6C68          LTA     IPILOT
0124 8084          MPYK    PLPF11
0125 4774          IN      TEMP1,7
0126 4F68          OUT     IPILOT,7
0127 6C74          LTA     TEMP1
0128 8081          MPYK    PLPF12
0129 4768          IN      IPILOT,7
012A 4F74          OUT     TEMP1,7
012B 6C68          LTA     IPILOT
012C 8031          MPYK    PLPF13
012D 4774          IN      TEMP1,7
012E 4F68          OUT     IPILOT,7
012F 6C74          LTA     TEMP1
0130 9FC7          MPYK    PLPF14
0131 4768          IN      IPILOT,7
0132 4F74          OUT     TEMP1,7
0133 6C68          LTA     IPILOT
0134 9F4D          MPYK    PLPF15
0135 4774          IN      TEMP1,7
0136 4F68          OUT     IPILOT,7
0137 6C74          LTA     TEMP1
0138 9ECD          MPYK    PLPF16
0139 4768          IN      IPILOT,7
013A 4F74          OUT     TEMP1,7
013B 6C68          LTA     IPILOT
013C 9E57          MPYK    PLPF17
013D 4774          IN      TEMP1,7
013E 4F68          OUT     IPILOT,7
```

```
013F 6C74   LTA    TEMP1
0140 9DF8   MPYK   PLPF18
0141 4768   IN     IPILOT,7
0142 4F74   OUT    TEMP1,7
0143 6C68   LTA    IPILOT
0144 9DCA   MPYK   PLPF19
0145 4774   IN     TEMP1,7
0146 4F68   OUT    IPILOT,7
0147 6C74   LTA    TEMP1
0148 9D04   MPYK   PLPF20
0149 4768   IN     IPILOT,7
014A 4F74   OUT    TEMP1,7
014B 6C68   LTA    IPILOT
014C 9E27   MPYK   PLPF21
014D 4774   IN     TEMP1,7
014E 4F68   OUT    IPILOT,7
014F 6C74   LTA    TEMP1
0150 9ECA   MPYK   PLPF22
0151 4768   IN     IPILOT,7
0152 4F74   OUT    TEMP1,7
0153 6C68   LTA    IPILOT
0154 9FC1   MPYK   PLPF23
0155 4774   IN     TEMP1,7
0156 4F68   OUT    IPILOT,7
0157 6C74   LTA    TEMP1
0158 8106   MPYK   PLPF24
0159 4768   IN     IPILOT,7
015A 4F74   OUT    TEMP1,7
015B 6C68   LTA    IPILOT
015C 828E   MPYK   PLPF25
015D 4774   IN     TEMP1,7
015E 4F68   OUT    IPILOT,7
015F 6C74   LTA    TEMP1
0160 8446   MPYK   PLPF26
0161 4768   IN     IPILOT,7
0162 4F74   OUT    TEMP1,7
0163 6C68   LTA    IPILOT
0164 8617   MPYK   PLPF27
0165 4774   IN     TEMP1,7
0166 4F68   OUT    IPILOT,7
0167 6C74   LTA    TEMP1
0168 87E5   MPYK   PLPF28
0169 4768   IN     IPILOT,7
016A 4F74   OUT    TEMP1,7
016B 6C68   LTA    IPILOT
016C 8394   MPYK   PLPF29
016D 4774   IN     TEMP1,7
016E 4F68   OUT    IPILOT,7
016F 6C74   LTA    TEMP1
0170 8806   MPYK   PLPF30
0171 4768   IN     IPILOT,7
0172 4F74   OUT    TEMP1,7
0173 6C68   LTA    IPILOT
0174 8C23   MPYK   PLPF31
0175 4774   IN     TEMP1,7
0176 4F68   OUT    IPILOT,7
0177 6C74   LTA    TEMP1
```

```
0178 8CC6    MPYK   PLPF32
0179 4768    IN     IPILOT,7
017A 4F74    OUT    TEMP1,7
017B 6C68    LTA    IPILOT
017C 8D13    MPYK   PLPF33
017D 4774    IN     TEMP1,7
017E 4F68    OUT    IPILOT,7
017F 6C74    LTA    TEMP1
0180 8CD6    MPYK   PLPF32
0181 4768    IN     IPILOT,7
0182 4F74    OUT    TEMP1,7
0183 6C68    LTA    IPILOT
0184 8C23    MPYK   PLPF31
0185 4774    IN     TEMP1,7
0186 4F68    OUT    IPILOT,7
0187 6C74    LTA    TEMP1
0188 8B06    MPYK   PLPF30
0189 4768    IN     IPILOT,7
018A 4F74    OUT    TEMP1,7
018B 6C68    LTA    IPILOT
018C 8994    MPYK   PLPF29
018D 4774    IN     TEMP1,7
018E 4F68    OUT    IPILOT,7
018F 6C74    LTA    TEMP1
0190 87E5    MPYK   PLPF28
0191 4768    IN     IPILOT,7
0192 4F74    OUT    TEMP1,7
0193 6C68    LTA    IPILOT
0194 8617    MPYK   PLPF27
0195 4774    IN     TEMP1,7
0196 4F68    OUT    IPILOT,7
0197 6C74    LTA    TEMP1
0198 8446    MPYK   PLPF26
0199 4768    IN     IPILOT,7
019A 4F74    OUT    TEMP1,7
019B 6C68    LTA    IPILOT
019C 828E    MPYK   PLPF25
019D 4774    IN     TEMP1,7
019E 4F68    OUT    IPILOT,7
019F 6C74    LTA    TEMP1
01A0 8106    MPYK   PLPF24
01A1 4768    IN     IPILOT,7
01A2 4F74    OUT    TEMP1,7
01A3 6C68    LTA    IPILOT
01A4 9FC1    MPYK   PLPF23
01A5 4774    IN     TEMP1,7
01A6 4F68    OUT    IPILOT,7
01A7 6C74    LTA    TEMP1
01A8 9ECA    MPYK   PLPF22
01A9 4768    IN     IPILOT,7
01AA 4F74    OUT    TEMP1,7
01AB 6C68    LTA    IPILOT
01AC 9E27    MPYK   PLPF21
01AD 4774    IN     TEMP1,7
01AE 4F68    OUT    IPILOT,7
01AF 6C74    LTA    TEMP1
01B0 9DD4    MPYK   PLPF20
```

```
01B1 4768        IN      IPILOT,7
01B2 4F74        OUT     TEMP1,7
01B3 6C68        LTA     IPILOT
01B4 9DCA        MPYK    PLPF19
01B5 4774        IN      TEMP1,7
01B6 4F68        OUT     IPILOT,7
01B7 6C74        LTA     TEMP1
01B8 9DFB        MPYK    PLPF18
01B9 4768        IN      IPILOT,7
01BA 4F74        OUT     TEMP1,7
01BB 6C68        LTA     IPILOT
01BC 9E57        MPYK    PLPF17
01BD 4774        IN      TEMP1,7
01BE 4F68        OUT     IPILOT,7
01BF 6C74        LTA     TEMP1
01C0 9ECD        MPYK    PLPF16
01C1 4768        IN      IPILOT,7
01C2 4F74        OUT     TEMP1,7
01C3 6C68        LTA     IPILOT
01C4 9F4D        MPYK    PLPF15
01C5 4774        IN      TEMP1,7
01C6 4F68        OUT     IPILOT,7
01C7 6C74        LTA     TEMP1
01C8 9FC7        MPYK    PLPF14
01C9 4768        IN      IPILOT,7
01CA 4F74        OUT     TEMP1,7
01CB 6C68        LTA     IPILOT
01CC 8031        MPYK    PLPF13
01CD 4774        IN      TEMP1,7
01CE 4F68        OUT     IPILOT,7
01CF 6C74        LTA     TEMP1
01D0 8081        MPYK    PLPF12
01D1 4768        IN      IPILOT,7
01D2 4F74        OUT     TEMP1,7
01D3 6C68        LTA     IPILOT
01D4 80B4        MPYK    PLPF11
01D5 4768        IN      IPILOT,7
01D6 4F68        OUT     IPILOT,7
01D7 6C74        LTA     TEMP1
01D8 80C3        MPYK    PLPF10
01D9 4768        IN      IPILOT,7
01DA 4F68        OUT     IPILOT,7
01DB 6C74        LTA     TEMP1
01DC 80C8        MPYK    PLPF9
01DD 4768        IN      IPILOT,7
01DE 4F68        OUT     IPILOT,7
01DF 6C74        LTA     TEMP1
01E0 80B0        MPYK    PLPF8
01E1 4768        IN      IPILOT,7
01E2 4F68        OUT     IPILOT,7
01E3 6C74        LTA     TEMP1
01E4 8083        MPYK    PLPF7
01E5 4768        IN      IPILOT,7
01E6 4F68        OUT     IPILOT,7
01E7 6C74        LTA     TEMP1
01E8 8060        MPYK    PLPF6
01E9 4768        IN      IPILOT,7
```

```
01EA 4F68          OUT     IPILOT,7
01EB 6C74          LTA     TEMP1
01EC 8036          MPYK    PLPF5
01ED 4768          IN      IPILOT,7
01EE 4F68          OUT     IPILOT,7
01EF 6C74          LTA     TEMP1
01F0 8011          MPYK    PLPF4
01F1 4768          IN      IPILOT,7
01F2 4F68          OUT     IPILOT,7
01F3 6C74          LTA     TEMP1
01F4 9FF6          MPYK    PLPF3
01F5 4768          IN      IPILOT,7
01F6 4F68          OUT     IPILOT,7
01F7 6C74          LTA     TEMP1
01F8 9FE6          MPYK    PLPF2
01F9 4768          IN      IPILOT,7
01FA 4F68          OUT     IPILOT,7
01FB 6C74          LTA     TEMP1
01FC 9F34          MPYK    PLPF1
01FD 7F8F          APAC
01FE 5968          SACH    IPILOT,1
              *
              *    Q Pilot LPF Code
              *
01FF 7F89          ZAC
0200 6A6B          LT      QPILOT
0201 9F34          MPYK    PLPF1
0202 4774          IN      TEMP1,7
0203 4F6B          OUT     QPILOT,7
0204 6C74          LTA     TEMP1
0205 9FE6          MPYK    PLPF2
0206 476B          IN      QPILOT,7
0207 4F74          OUT     TEMP1,7
0208 6C6B          LTA     QPILOT
0209 9FF6          MPYK    PLPF3
020A 4774          IN      TEMP1,7
020B 4F6B          OUT     QPILOT,7
020C 6C74          LTA     TEMP1
020D 8011          MPYK    PLPF4
020E 476B          IN      QPILOT,7
020F 4F74          OUT     TEMP1,7
0210 6C6B          LTA     QPILOT
0211 8036          MPYK    PLPF5
0212 4774          IN      TEMP1,7
0213 4F6B          OUT     QPILOT,7
0214 6C74          LTA     TEMP1
0215 8060          MPYK    PLPF6
0216 476B          IN      QPILOT,7
0217 4F74          OUT     TEMP1,7
0218 6C6B          LTA     QPILOT
0219 808B          MPYK    PLPF7
021A 4774          IN      TEMP1,7
021B 4F6B          OUT     QPILOT,7
021C 6C74          LTA     TEMP1
021D 8080          MPYK    PLPF8
021E 476B          IN      QPILOT,7
021F 4F74          OUT     TEMP1,7
```

```
0220 6C6B    LTA    QPILOT
0221 80C8    MPYK   PLPF9
0222 4774    IN     TEMP1,7
0223 4F6B    OUT    QPILOT,7
0224 6C74    LTA    TEMP1
0225 80CB    MPYK   PLPF10
0226 476B    IN     QPILOT,7
0227 4F74    OUT    TEMP1,7
0228 6C6B    LTA    QPILOT
0229 80B4    MPYK   PLPF11
022A 4774    IN     TEMP1,7
022B 4F6B    OUT    QPILOT,7
022C 6C74    LTA    TEMP1
022D 8081    MPYK   PLPF12
022E 476B    IN     QPILOT,7
022F 4F74    OUT    TEMP1,7
0230 6C6B    LTA    QPILOT
0231 8031    MPYK   PLPF13
0232 4774    IN     TEMP1,7
0233 4F6B    OUT    QPILOT,7
0234 6C74    LTA    TEMP1
0235 9FC7    MPYK   PLPF14
0236 476B    IN     QPILOT,7
0237 4F74    OUT    TEMP1,7
0238 6C6B    LTA    QPILOT
0239 9F4D    MPYK   PLPF15
023A 4774    IN     TEMP1,7
023B 4F6B    OUT    QPILOT,7
023C 6C74    LTA    TEMP1
023D 9ECD    MPYK   PLPF16
023E 476B    IN     QPILOT,7
023F 4F74    OUT    TEMP1,7
0240 6C6B    LTA    QPILOT
0241 9E57    MPYK   PLPF17
0242 4774    IN     TEMP1,7
0243 4F6B    OUT    QPILOT,7
0244 6C74    LTA    TEMP1
0245 9DFB    MPYK   PLPF18
0246 476B    IN     QPILOT,7
0247 4F74    OUT    TEMP1,7
0248 6C6B    LTA    QPILOT
0249 9DCA    MPYK   PLPF19
024A 4774    IN     TEMP1,7
024B 4F6B    OUT    QPILOT,7
024C 6C74    LTA    TEMP1
024D 9DD4    MPYK   PLPF20
024E 476B    IN     QPILOT,7
024F 4F74    OUT    TEMP1,7
0250 6C6B    LTA    QPILOT
0251 9E27    MPYK   PLPF21
0252 4774    IN     TEMP1,7
0253 4F6B    OUT    QPILOT,7
0254 6C74    LTA    TEMP1
0255 9ECA    MPYK   PLPF22
0256 476B    IN     QPILOT,7
0257 4F74    OUT    TEMP1,7
0258 6C6B    LTA    QPILOT
```

```
0259 9FC1   MPYK  PLPF23
025A 4774   IN    TEMP1,7
025B 4F6B   OUT   QPILOT,7
025C 6C74   LTA   TEMP1
025D 8106   MPYK  PLPF24
025E 476B   IN    QPILOT,7
025F 4F74   OUT   TEMP1,7
0260 6C6B   LTA   QPILOT
0261 828E   MPYK  PLPF25
0262 4774   IN    TEMP1,7
0263 4F6B   OUT   QPILOT,7
0264 6C74   LTA   TEMP1
0265 8446   MPYK  PLPF26
0266 476B   IN    QPILOT,7
0267 4F74   OUT   TEMP1,7
0268 6C6B   LTA   QPILOT
0269 8617   MPYK  PLPF27
026A 4774   IN    TEMP1,7
026B 4F6B   OUT   QPILOT,7
026C 6C74   LTA   TEMP1
026D 87E5   MPYK  PLPF28
026E 476B   IN    QPILOT,7
026F 4F74   OUT   TEMP1,7
0270 6C6B   LTA   QPILOT
0271 8994   MPYK  PLPF29
0272 4774   IN    TEMP1,7
0273 4F6B   OUT   QPILOT,7
0274 6C74   LTA   TEMP1
0275 8806   MPYK  PLPF30
0276 476B   IN    QPILOT,7
0277 4F74   OUT   TEMP1,7
0278 6C6B   LTA   QPILOT
0279 8C23   MPYK  PLPF31
027A 4774   IN    TEMP1,7
027B 4F6B   OUT   QPILOT,7
027C 6C74   LTA   TEMP1
027D 8CD6   MPYK  PLPF32
027E 476B   IN    QPILOT,7
027F 4F74   OUT   TEMP1,7
0280 6C6B   LTA   QPILOT
0281 8D13   MPYK  PLPF33
0282 4774   IN    TEMP1,7
0283 4F6B   OUT   QPILOT,7
0284 6C74   LTA   TEMP1
0285 8CD6   MPYK  PLPF32
0286 476B   IN    QPILOT,7
0287 4F74   OUT   TEMP1,7
0288 6C6B   LTA   QPILOT
0289 8C23   MPYK  PLPF31
028A 4774   IN    TEMP1,7
028B 4F6B   OUT   QPILOT,7
028C 6C74   LTA   TEMP1
028D 8806   MPYK  PLPF30
028E 476B   IN    QPILOT,7
028F 4F74   OUT   TEMP1,7
0290 6C6B   LTA   QPILOT
0291 8994   MPYK  PLPF29
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0292 4774      IN      TEMP1,7
0293 4F6B      OUT     QPILOT,7
0294 6C74      LTA     TEMP1
0295 87E5      MPYK    PLPF28
0296 476B      IN      QPILOT,7
0297 4F74      OUT     TEMP1,7
0298 6C6B      LTA     QPILOT
0299 8617      MPYK    PLPF27
029A 4774      IN      TEMP1,7
029B 4F6B      OUT     QPILOT,7
029C 6C74      LTA     TEMP1
029D 8446      MPYK    PLPF26
029E 476B      IN      QPILOT,7
029F 4F74      OUT     TEMP1,7
02A0 6C6B      LTA     QPILOT
02A1 828E      MPYK    PLPF25
02A2 4774      IN      TEMP1,7
02A3 4F6B      OUT     QPILOT,7
02A4 6C74      LTA     TEMP1
02A5 8106      MPYK    PLPF24
02A6 476B      IN      QPILOT,7
02A7 4F74      OUT     TEMP1,7
02A8 6C6B      LTA     QPILOT
02A9 9FC1      MPYK    PLPF23
02AA 4774      IN      TEMP1,7
02AB 4F6B      OUT     QPILOT,7
02AC 6C74      LTA     TEMP1
02AD 9ECA      MPYK    PLPF22
02AE 476B      IN      QPILOT,7
02AF 4F74      OUT     TEMP1,7
02B0 6C6B      LTA     QPILOT
02B1 9E27      MPYK    PLPF21
02B2 4774      IN      TEMP1,7
02B3 4F6B      OUT     QPILOT,7
02B4 6C74      LTA     TEMP1
02B5 9DD4      MPYK    PLPF20
02B6 476B      IN      QPILOT,7
02B7 4F74      OUT     TEMP1,7
02B8 6C6B      LTA     QPILOT
02B9 9DCA      MPYK    PLPF19
02BA 4774      IN      TEMP1,7
02BB 4F6B      OUT     QPILOT,7
02BC 6C74      LTA     TEMP1
02BD 9DFB      MPYK    PLPF18
02BE 476B      IN      QPILOT,7
02BF 4F74      OUT     TEMP1,7
02C0 6C6B      LTA     QPILOT
02C1 9E57      MPYK    PLPF17
02C2 4774      IN      TEMP1,7
02C3 4F6B      OUT     QPILOT,7
02C4 6C74      LTA     TEMP1
02C5 9ECD      MPYK    PLPF16
02C6 476B      IN      QPILOT,7
02C7 4F74      OUT     TEMP1,7
02C8 6C6B      LTA     QPILOT
02C9 9F4D      MPYK    PLPF15
02CA 4774      IN      TEMP1,7
```

```
02CB 4F6B    OUT   QPILOT,7
02CC 6C74    LTA   TEMP1
02CD 9FC7    MPYK  PLPF14
02CE 476B    IN    QPILOT,7
02CF 4F74    OUT   TEMP1,7
02D0 6C6B    LTA   QPILOT
02D1 8031    MPYK  PLPF13
02D2 4774    IN    TEMP1,7
02D3 4F6B    OUT   QPILOT,7
02D4 6C74    LTA   TEMP1
02D5 8081    MPYK  PLPF12
02D6 476B    IN    QPILOT,7
02D7 4F74    OUT   TEMP1,7
02D8 6C6B    LTA   QPILOT
02D9 8084    MPYK  PLPF11
02DA 4768    IN    IPILOT,7
02DB 4F68    OUT   IPILOT,7
02DC 6C74    LTA   TEMP1
02DD 80C9    MPYK  PLPF10
02DE 4768    IN    IPILOT,7
02DF 4F68    OUT   IPILOT,7
02E0 6C74    LTA   TEMP1
02E1 80C8    MPYK  PLPF9
02E2 4768    IN    IPILOT,7
02E3 4F68    OUT   IPILOT,7
02E4 6C74    LTA   TEMP1
02E5 80B0    MPYK  PLPF8
02E6 4768    IN    IPILOT,7
02E7 4F68    OUT   IPILOT,7
02E8 6C74    LTA   TEMP1
02E9 808B    MPYK  PLPF7
02EA 4768    IN    IPILOT,7
02EB 4F68    OUT   IPILOT,7
02EC 6C74    LTA   TEMP1
02ED 8060    MPYK  PLPF6
02EE 4768    IN    IPILOT,7
02EF 4F68    OUT   IPILOT,7
02F0 6C74    LTA   TEMP1
02F1 8036    MPYK  PLPF5
02F2 4768    IN    IPILOT,7
02F3 4F68    OUT   IPILOT,7
02F4 6C74    LTA   TEMP1
02F5 8011    MPYK  PLPF4
02F6 4768    IN    IPILOT,7
02F7 4F68    OUT   IPILOT,7
02F8 6C74    LTA   TEMP1
02F9 9FF6    MPYK  PLPF3
02FA 4768    IN    IPILOT,7
02FB 4F68    OUT   IPILOT,7
02FC 6C74    LTA   TEMP1
02FD 9FE6    MPYK  PLPF2
02FE 4768    IN    IPILOT,7
02FF 4F68    OUT   IPILOT,7
0300 6C74    LTA   TEMP1
0301 9F34    MPYK  PLPF1
0302 7F8F    APAC
0303 596B    SACH  QPILOT,1
```

```
                *
                *       Sine and Cosine Calculation
                *
                *       The values for sin & cos are
                *       stored for the region 0 - pi/4
                *       in 128 locations each; the pilot
                *       samples are first stripped for
                *       sign and then compared to determine
                *       the octant before table lookup;
                *       the sign is then re-appended after
                *       the values have been determined;
                *
0304 656B               ZALH    QPILOT          }
0305 587C               SACH    QSIGN           } Strip Q pilot sign.
0306 7F88               ABS                     }
0307 5874               SACH    TEMP1
0308 6568               ZALH    IPILOT          }
0309 587B               SACH    ISIGN           } Strip I pilot sign.
030A 7F88               ABS                     }
030B 5875               SACH    TEMP2
030C 6274               SUBH    TEMP1           } Compare magnitudes to
030D FA00               BLZ     IOVERQ          } determine octant.
030E 0342
                *
030F 6574       QOVERI  ZALH    TEMP1           } Check if denominator
0310 FE00               BNZ     DIVQOI          } equals zero.
0311 0319
0312 7F89               ZAC
0313 5061               SACL    COS
0314 7E01               LACK    >1
0315 007A               ADD     SNOFST
0316 675E               TBLR    SIN
0317 F900               B       FIXSGN
0318 032F
                *
                *       N.B. : NOP's are inserted after each SUBC
                *              instruction because the instruction
                *              following a SUBC may not use the
                *              accumulator.
                *
0319 6475       DIVQOI  SUBC    TEMP2           } 7 Bit Fractional
031A 7F80               NOP
031B 6475               SUBC    TEMP2           }     Divide
031C 7F80               NOP
031D 6475               SUBC    TEMP2           }       .
031E 7F80               NOP
031F 6475               SUBC    TEMP2           }       .
0320 7F80               NOP
0321 6475               SUBC    TEMP2           }       .
0322 7F80               NOP
0323 6475               SUBC    TEMP2           }       .
0324 7F80               NOP
0325 6475               SUBC    TEMP2           } End of Divide
0326 7F80               NOP
0327 5075               SACL    TEMP2
0328 2175               LAC     TEMP2,1
0329 007A               ADD     SNOFST
```

```
0032A 675E          TBLR    SIN             ) Read sine value.
0328 5075           SACL    TEMP2
032C 7E01           LACK    1               ) Increment lookup address.
032D 0075           ADD     TEMP2
032E 6761           TBLR    COS             ) Read cosine value.
032F 657B   FIXSGN  ZALH    ISIGN
0330 FA00           BLZ     SUBI1           )
0331 0337
0332 657C           ZALH    QSIGN           ) Re-append Sign.
0333 FA00           BLZ     SUBQ1           )
0334 033E
0335 F900           B       STEPS
0336 0373
0337 1061   SUBI1   SUB     COS
0338 5061           SACL    COS
0339 657C           ZALH    QSIGN
033A FA00           BLZ     SUBQ1
033B 033E
033C F900           B       STEPS
033D 0373
033E 105E   SUBQ1   SUB     SIN
033F 505E           SACL    SIN
0340 F900           B       STEPS
0341 0373
            *
0342 6575   IOVERQ  ZALH    TEMP2           ) Check if denominator
0343 FE00           BNZ     DIVIDQ          ) equals zero.
0344 034C
0345 7F89           ZAC
0346 505E           SACL    SIN
0347 7E01           LACK    >1
0348 007A           ADD     SNOFST
0349 6761           TBLR    COS
034A F900           B       SGNFIX
034B 0362
            *
            *       N.B. : NOP's are inserted after each SUBC
            *              instruction because the instruction
            *              following a SUBC may not use the
            *              accumulator.
            *
034C 6474   DIVIDQ  SUBC     TEMP1          ) 7 Bit Fractional
034D 7F80           NOP
034E 6474           SUBC     TEMP1          )     Divide
034F 7F80           NOP
0350 6474           SUBC     TEMP1          )        .
0351 7F80           NOP
0352 6474           SUBC     TEMP1          )        .
0353 7F80           NOP
0354 6474           SUBC     TEMP1          )        .
0355 7F80           NOP
0356 6474           SUBC     TEMP1          )        .
0357 7F80           NOP
0358 6474           SUBC     TEMP1          ) End of Divide
0359 7F80           NOP
035A 5074           SACL     TEMP1
035B 2174           LAC      TEMP1,1
```

```
0350 007A          ADD    SNOFST
035D 6761          TBLR   COS        ) Read cosine value.
035E 5075          SACL   TEMP2
035F 7E01          LACK   1          ) Increment lookup address.
0360 0075          ADD    TEMP2
0361 675E          TBLR   SIN        ) Read sine value.
0362 657B  SGNFIX  ZALH   ISIGN
0363 FA00          BLZ    SUBI2      )
0364 036A
0365 657C          ZALH   QSIGN      ) Re-append Sign.
0366 FA00          BLZ    SUBQ2      )
0367 0371
0368 F900          B      STEPS
0369 0373
036A 1061  SUBI2   SUB    COS
036B 5061          SACL   COS
036C 657C          ZALH   QSIGN
036D FA00          BLZ    SUBQ2
036E 0371
036F F900          B      STEPS
0370 0373
0371 105E  SUBQ2   SUB    SIN
0372 505E          SACL   SIN
           *
           *       Calculate Pilot, Cosine
           *       and Sine Step Sizes.
           *
0373 6562  STEPS   ZALH   PRECOS
0374 6261          SUBH   COS
0375 587B          SACH   ISIGN,0
0376 7F88          ABS
0377 5874          SACH   TEMP1,0
0378 6A74          LT     TEMP1
           *
0379 8333          MPYK   >333       ) Multiply by 1/5.
037A 7F8E          PAC
037B 0B72          ADD    ONE,11     ) Round off result.
           *
037C 5066          SACL   COSSTP
037D 657B          ZALH   ISIGN
037E FD00          BGEZ   NEXT1
037F 0382
0380 1066          SUB    COSSTP     ) Correct for negative
0381 5066          SACL   COSSTP     ) result.
           *
0382 655F  NEXT1   ZALH   PRESIN
0383 625E          SUBH   SIN
0384 587C          SACH   QSIGN,0
0385 7F88          ABS
0386 5874          SACH   TEMP1,0
0387 6A74          LT     TEMP1
           *
0388 8333          MPYK   >333       ) Multiply by 1/5.
0389 7F8E          PAC
038A 0B72          ADD    ONE,11     ) Round off result.
           *
038B 5064          SACL   SINSTP
```

```
038C 657C          ZALH    QSIGN
038D F000          BGEZ    NEXT2
038E 0391
038F 1064          SUB     SINSTP          } Correct for negative
0390 5064          SACL    SINSTP          } result.
                 *
0391 6569  NEXT2   ZALH    OLDIP
0392 6268          SUBH    IPILOT
0393 587B          SACH    ISIGN,0
0394 7F88          ABS
0395 5874          SACH    TEMP1,0
0396 6A74          LT      TEMP1
                 *
0397 8333          MPYK    >333            } Multiply by 1/5.
0398 7F8E          PAC
0399 0872          ADD     ONE,11          } Round off result.
                 *
039A 506E          SACL    IPSTP
039B 657B          ZALH    ISIGN
039C F000          BGEZ    NEXT3
039D 03A0
039E 106E          SUB     IPSTP           } Correct for negative
039F 506E          SACL    IPSTP           } result.
                 *
03A0 656C  NEXT3   ZALH    OLDQP
03A1 626B          SUBH    QPILOT
03A2 587C          SACH    QSIGN,0
03A3 7F88          ABS
03A4 5874          SACH    TEMP1
03A5 6A74          LT      TEMP1
                 *
03A6 8333          MPYK    >333            } Multiply by 1/5.
03A7 7F8E          PAC
03A8 0872          ADD     ONE,11          } Round off result.
                 *
03A9 5070          SACL    QPSTP
03AA 657C          ZALH    QSIGN
                 *
03AB 4D72          OUT     ONE,5           } Reset ram counters.
                 *
03AC F000          BGEZ    BKGRND
03AD 00F7
03AE 1070          SUB     QPSTP           } Correct for negative
03AF 5070          SACL    QPSTP           } result.
                 *
03B0 F900          B       BKGRND
03B1 00F7
                 *
                 *         Reset Initialization Routine
                 *
03B2 6E00  RESET   LDPK    0               } Point to data page 0.
03B3 7F8B          SOVM                    } Set overflow mode.
                 *
03B4 4000          IN      >0,0            } Clear interrupt pin.
03B5 4D00          OUT     >0,5            } Reset ram counters.
                 *
03B6 7E01          LACK    1
```

```
0387 5000           SACL    >0
0388 6A00           LT      >0
                *
0389 6880           LARP    0
038A 707F           LARK    0,>7F
038B 7F89           ZAC                    } 
038C 5088   CLRRAM  SACL    *              } Clear internal ram.
038D F400           BANZ    CLRRAM         }
038E 038C
                *
038F 7054           LARK    0,IBUFF        } Initialize auxiliary
03C0 7159           LARK    1,QBUFF        } register pointers.
                *
03C1 7E01           LACK    1              }
03C2 5072           SACL    ONE            }
03C3 8500           MPYK    OFFSET         } Store constants in ram.
03C4 7F8E           PAC                    }
03C5 507A           SACL    CSOFST         }
                *
03C6 F900           B       BKGRND         } Branch to waiting state.
03C7 00F7
                *
                *
                *       Sine and Cosine Table Lookup Values
                *
                *
0500                AORG    >500
                *
0500 0000           DATA    0,32767,258,32767,516,32764
0501 7FFF
0502 0102
0503 7FFF
0504 0204
0505 7FFC
0506 0306           DATA    774,32759,1032,32752,1289,32743
0507 7FF7
0508 0408
0509 7FF0
050A 0509
050B 7FE7
050C 060A           DATA    1546,32731,1803,32718,2060,32703
050D 7FDB
050E 070B
050F 7FCE
0510 080C
0511 7FBF
0512 090C           DATA    2316,32686,2572,32667,2828,32646
0513 7FAE
0514 0A0C
0515 7F9B
0516 0B0C
0517 7F86
0518 0C0A           DATA    3082,32623,3337,32598,3590,32571
0519 7F6F
051A 0D09
051B 7F56
051C 0E06
```

```
      051C 7F3B
1318  051E 0F04          DATA     3844,32542,4096,32511,4347,32478
      051F 7F1E
      0520 1000
      0521 7EFF
      0522 10FB
      0523 7EDE
1319  0524 11F6          DATA     4598,32444,4848,32407,5097,32369
      0525 7EBC
      0526 12F0
      0527 7E97
      0528 13E9
      0529 7E71
1320  052A 14E2          DATA     5346,32329,5593,32287,5839,32244
      052B 7E49
      052C 15D9
      052D 7E1F
      052E 16CF
      052F 7DF4
1321  0530 17C5          DATA     6085,32198,6329,32151,6572,32102
      0531 7DC6
      0532 18B9
      0533 7D97
      0534 19AC
      0535 7D66
1322  0536 1A9E          DATA     6814,32052,7055,32000,7295,31946
      0537 7D34
      0538 1B8F
      0539 7D00
      053A 1C7F
      053B 7CCA
1323  053C 1D6D          DATA     7533,31890,7770,31833,8006,31775
      053D 7C92
      053E 1E5A
      053F 7C59
      0540 1F46
      0541 7C1F
1324  0542 2031          DATA     8241,31715,8474,31653,8706,31590
      0543 7BE3
      0544 211A
      0545 7BA5
      0546 2202
      0547 7B66
1325  0548 22E8          DATA     8936,31526,9166,31460,9393,31393
      0549 7B26
      054A 23CE
      054B 7AE4
      054C 2481
      054D 7AA1
1326  054E 2593          DATA     9619,31324,9844,31254,10067,31183
      054F 7A5C
      0550 2674
      0551 7A16
      0552 2753
      0553 79CF
1327  0554 2831          DATA     10289,31111,10509,31037,10727,30962
      0555 7987
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      0556 290D
      0557 793D
      0558 29E7
      0559 78F2
 1328 055A 2AC0          DATA     10944,30886,11159,30809,11373,30731
      055B 78A6
      055C 2B97
      055D 7859
      055E 2C6D
      055F 780B
 1329 0560 2D41          DATA     11585,30652,11795,30571,12004,30490
      0561 77EC
      0562 2E13
      0563 776B
      0564 2EE4
      0565 771A
 1330 0566 2FB3          DATA     12211,30408,12416,30325,12620,30240
      0567 76C8
      0568 3080
      0569 7675
      056A 314C
      056B 7620
 1331 056C 3216          DATA     12822,30155,13022,30069,13221,29983
      056D 75C8
      056E 32DE
      056F 7575
      0570 33A5
      0571 751F
 1332 0572 3469          DATA     13417,29895,13613,29807,13806,29718
      0573 74C7
      0574 352D
      0575 746F
      0576 35EE
      0577 7416
 1333 0578 36AD          DATA     13997,29628,14187,29537,14375,29446
      0579 73BC
      057A 376B
      057B 7361
      057C 3827
      057D 7306
 1334 057E 38E2          DATA     14562,29355,14746,29262,14929,29169
      057F 72AB
      0580 399A
      0581 724E
      0582 3A51
      0583 71F1
 1335 0584 3B06          DATA     15110,29076,15290,28982,15467,28888
      0585 7194
      0586 3BBA
      0587 7136
      0588 3C6B
      0589 70D8
 1336 058A 3D1B          DATA     15643,28793,15818,28698,15990,28602
      058B 7079
      058C 3DCA
      058D 701A
      058E 3E76
```

```
     058F 6FBA
1337 0590 3F21          DATA     16161,28506,16330,28409,16497,28312
     0591 6F5A
     0592 3FCA
     0593 6EF9
     0594 4071
     0595 6E98
1338 0596 4117          DATA     16663,28215,16826,28118,16989,28020
     0597 6E37
     0598 41BA
     0599 6DD6
     059A 425D
     059B 6D74
1339 059C 42FD          DATA     17149,27922,17308,27824,17465,27726
     059D 6D12
     059E 439C
     059F 6CB0
     05A0 4439
     05A1 6C4E
1340 05A2 44D4          DATA     17620,27627,17774,27528,17926,27430
     05A3 6BEB
     05A4 456E
     05A5 6B88
     05A6 4606
     05A7 6B26
1341 05A8 469D          DATA     18077,27331,18226,27232,18373,27132
     05A9 6AC3
     05AA 4732
     05AB 6A60
     05AC 47C5
     05AD 69FC
1342 05AE 4857          DATA     18519,27033,18663,26934,18805,26835
     05AF 6999
     05B0 48E7
     05B1 6936
     05B2 4975
     05B3 68D3
1343 05B4 4A02          DATA     18946,26735,19086,26636,19223,26537
     05B5 686F
     05B6 4A8E
     05B7 680C
     05B8 4B17
     05B9 67A9
1344 05BA 4BA0          DATA     19360,26438,19494,26338,19628,26239
     05BB 6746
     05BC 4C26
     05BD 66E2
     05BE 4CAC
     05BF 667F
1345 05C0 4D2F          DATA     19759,26140,19890,26041,20018,25942
     05C1 661C
     05C2 4DB2
     05C3 65B9
     05C4 4E32
     05C5 6556
1346 05C6 4EB2          DATA     20146,25844,20272,25745,20396,25647
     05C7 64F4
```

```
      05C8 4F30
      05C9 6491
      05CA 4FAC
      05CB 642F
 1347 05CC 5027          DATA      20519,25548,20641,25450,20761,25352
      05CD 63CC
      05CE 50A1
      05CF 636A
      05D0 5119
      05D1 6308
 1348 05D2 5190          DATA      20880,25254,20997,25157,21113,25059
      05D3 62A6
      05D4 5205
      05D5 6245
      05D6 5279
      05D7 61E3
 1349 05D8 52EC          DATA      21228,24962,21341,24865,21453,24769
      05D9 6182
      05DA 5350
      05DB 6121
      05DC 53CD
      05DD 60C1
 1350 05DE 543C          DATA      21564,24672,21674,24576,21782,24480
      05DF 6060
      05E0 54AA
      05E1 6000
      05E2 5516
      05E3 5FA0
 1351 05E4 5581          DATA      21889,24385,21995,24290,22099,24195
      05E5 5F41
      05E6 55EB
      05E7 5EE2
      05E8 5653
      05E9 5E83
 1352 05EA 56BA          DATA      22202,24100,22304,24005,22405,23911
      05EB 5E24
      05EC 5720
      05ED 5DC5
      05EE 5785
      05EF 5D67
 1353 05F0 57E9          DATA      22505,23818,22603,23724,22701,23631
      05F1 5D0A
      05F2 584B
      05F3 5CAC
      05F4 58AD
      05F5 5C4F
 1354 05F6 5900          DATA      22797,23538,22892,23446,22986,23354
      05F7 5BF2
      05F8 596C
      05F9 5B96
      05FA 59CA
      05FB 5B3A
 1355 05FC 5A27          DATA      23079,23262,23170,23170
      05FD 5ADE
      05FE 5A82
      05FF 5A82
 1356           *
```

```
                  *       Bootstrapping Routine for
                  •       Loading Program Code from
                  •       EPROM's to RAM.
                  *
0700                      AORG     >700
                  *
0700  7E01        BOOT    LACK     >1
0701  5000                SACL     >0
0702  6A00                LT       >0
0703  87FF                MPYK     >7FF
0704  7F8E                PAC
0705  670A        NOTDUN  TBLR     >A
0706  7D0A                TBLW     >A
0707  1000                SUB      >0
0708  FD00                BGEZ     NOTDUN
0709  0705
070A  83B2                MPYK     RESET
070B  7F8E                PAC
070C  500A                SACL     >A
070D  7E01                LACK     >1
070E  7D0A                TBLW     >A
070F  F900                B        RESET
0710  03B2
                  *
                          END
NO ERRORS, NO WARNINGS
```